国外工业控制与智能制造丛书

HZ BOOKS 华章教育

WILEY

自适应系统与机器智能

何海波 (Haibo He) 著

薛建儒 王晓峰 译

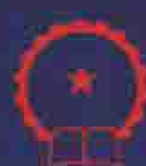
Self-adaptive Systems for Machine Intelligence



SELF-ADAPTIVE SYSTEMS FOR MACHINE INTELLIGENCE

HAIBO HE

WILEY



机械工业出版社
China Machine Press

机器智能研究是关于自适应系统的原理、基础和设计的研究，这种自适应系统能够学习、预测和优化，并通过与不确定的环境交互做出决策，从而完成系统目标。本书有助于对自适应智能系统的基本理解，促进读者向模拟某些类脑智能水平的长期目标前进，同时也使如今许多复杂系统的智能水平更接近现实。

本书分为以下4个主要部分

- 第一部分介绍了用于机器智能研究的自适应系统，给出了研究的意义以及传统计算机与类脑智能的主要区别。
- 第二部分重点讨论了机器智能研究的数据驱动方法，着重介绍了增量学习、不平衡学习和集成学习。
- 第三部分着重介绍机器智能研究的生物启发式方法，详细讨论了自适应动态规划、联想学习和序列学习。
- 第四部分简要介绍机器智能关键硬件的设计，如功耗、设计密度、内存和速度，目的是实现大规模、复杂的综合智能硬件系统。

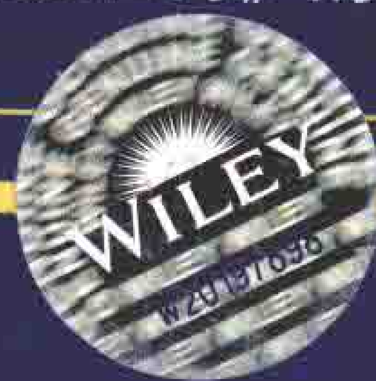
不同的应用问题（如模式识别、数据分类、自适应控制、图像恢复）显示了该系统的学习、预测和优化能力。本书提供的原理、体系结构、算法和案例研究，不但为机器智能研究提供了新的观点，而且为广泛的实际应用提供了新的技术和解决方案。书中讨论的所有问题都属于相关领域内具有重大挑战性的热门研究主题，这使得本书成为研究生激励自己向博士研究项目或大师级研究水平迈进的宝贵资源。本书也适用于计算智能/机器学习领域的学术研究人员和专业人员、对自适系统应感兴趣的工业研究人员和研发工程师，以及科学或工程专业的本科生参考。

作者简介

何海波（Haibo He）美国罗德岛大学的讲席教授（Robert Haas Endowed Professor）、智能计算与自适应系统实验室主任，主要从事智能计算、控制与优化、机器学习、大数据、网络安全、大规模复杂系统等方向的研究。

何教授曾在权威学术期刊和会议上发表论文200多篇，这些论文在专业领域产生了深远的影响。截止2016年4月，何博士的论文总引用次数达4100+，h-index 是27。2006年以来，他作为项目负责人承担科研项目经费超过600万美元，资助方包括美国国家自然科学基金委、美国国家航空航天局、美国海洋能源管理局，以及工业界。

何教授还担任多个国际会议的总主席、技术委员会主席等，也曾担任IEEE智能计算协会新兴技术委员会主席、IEEE智能计算协会神经网络技术委员会副主席、IEEE智能计算智能电网副主席等。目前担任期刊《IEEE 神经网络学习系统汇刊》的主编。





WILEY
www.wiley.com

投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

封面设计 · 锡彬

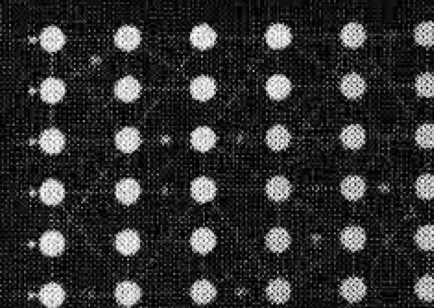
上架指导：智能系统

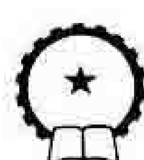

ISBN 978-7-111-54114-1
定价：59.00元

自适应系统与机器智能

何海波 (Haibo He) 著
薛建儒 王晓峰 译

Self-adaptive Systems for Machine Intelligence




机械工业出版社
China Machine Press

图书在版编目（CIP）数据

自适应系统与机器智能 / 何海波著；薛建儒，王晓峰译．—北京：机械工业出版社，2016.7
（国外工业控制与智能制造丛书）
书名原文：Self-Adaptive Systems for Machine Intelligence

ISBN 978-7-111-54114-1

I. 自…　II. ① 何…　② 薛…　③ 王…　III. 人工智能－自适应控制系统－研究　IV. TP18

中国版本图书馆 CIP 数据核字（2016）第 159828 号

本书版权登记号：图字：01-2014-4438

本书综合了多个领域的最新研究成果，阐述了机器智能的计算基础和方法论，强调自适应智能系统的“计算思维”能力的设计，主要讨论了机器智能的数据驱动与生物启发式两类方法，提出了在理解生物脑组织中神经信息处理的基本原理、机制的基础上，实现学习、记忆、预测和优化的通用机器智能方法。本书面向对机器智能领域感兴趣的研究人员和从业人员，目的是促进他们理解机器智能研究方面的自适应系统，并给出能够自适应学习信息、随着时间积累知识、调节行为来实现目标的模型与架构。本书所介绍的学习原则、体系结构、算法和实例研究，不仅有希望能为机器智能研究领域带来新见解，而且提供了潜在的技术和解决方案，从而使机器智能的智力水平在广泛的应用领域中更加接近现实。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）

责任编辑：张梦玲	责任校对：董纪丽
印　　刷：北京诚信伟业印刷有限公司	版　　次：2016 年 7 月第 1 版第 1 次印刷
开　　本：185mm×260mm　1/16	印　　张：12.75
书　　号：ISBN 978-7-111-54114-1	定　　价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991　88361066　　投稿热线：（010）88379604

购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，信息学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的信息产业发展迅猛，对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson、McGraw-Hill、Elsevier、John Wiley & Sons、CRC、Springer 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Alan V. Oppenheim Thomas L. Floyd、Charles K. Alexander、Behzad Razavi、John G. Proakis、Stephen Brown、Allan R. Hambley、Albert Malvino、Peter Wilson、H. Vincent Poor、Hassan K. Khalil、Gene F. Franklin、Rex Miller 等大师名家的经典教材，以“国外电子与电气技术丛书”和“国外工业控制与智能丛书”为系列出版，供读者学习、研究及珍藏。这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也越来越多被实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着电气与电子信息学科建设的不断完善和教材改革的逐渐深化，教育界对国外国外电气与电子信息教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

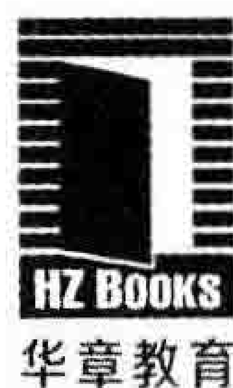
华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037



华章科技图书出版中心

译者序

由笔者组织本书的翻译纯属偶然。2014年的某日下午，机械工业出版社华章公司王颖女士专程来实验室讨论智能系统学科发展，后又致电委托评审这本计划引进的学术专著，再后又委托组织翻译。盛情难却，加之的确有义务推介人工智能研究领域的最新研究成果，于是在繁重的科研任务间隙，与西安理工大学王晓峰教授合作，历时半年完成了翻译工作。

付梓在即，回顾过去，发现这段时间正值人工智能50多年历史上发展最为迅速、令人激动的研究成果不断涌现的两年，互联网催生的海量数据与大幅度提升的计算及存储能力使得以传统神经网络为基础的深度学习，以网络搜索技术为引擎的群体智能，高度集成感知、交互与运动控制的无人系统等研究取得了令人瞩目的重要进展，彻底颠覆了过去人们对人工智能的观感和认识。

此外，人工智能在相继出现的智慧地球、感知中国、云计算、大数据、智能机器人等热点应用的驱动下，“以人为中心”“人在环路的智能计算”“把机器智能作为人类智能的有效扩展”“类脑计算”等观点成为人工智能领域的普遍共识，人工智能的研究范围因而不断拓宽，已涵盖了模式识别、智能信息处理、自然语言理解、知识工程与认知科学等多个重要分支，并在不断更新与扩展。人工智能现已成为专门研究与人的感知、思维、决策、问题求解和学习等相关的智能活动的自动化方法、智能信息处理技术、智能机器系统的新兴前沿学科。

毋庸置疑，当前和将来的人工智能已经从人如何适应机器发展到机器与人交互、理解人并更好地服务于人的阶段，尤其是人工智能与生物神经科学、人类心理学和脑科学等新兴学科的深度交叉，使得机器智能与人类智能的界限日益模糊。然而，正如郑南宁院士在“人工智能发展的下一步是什么”的学术报告中指出：实现人类水平的人工智能需要应对诸多巨大挑战，例如，如何让机器在没有人类教师的帮助下学习？如何让机器像人类一样感知和理解世界？如何让机器具有自我意识、情感以及反思自身处境与行为的能力？

本书围绕机器智能如何像生物智能一样自适应于环境这一科学问题，从数据驱动与生物启发两个层面出发，提出了在理解生物脑组织中神经信息处理的基本原理和机制的基础上，实现学习、记忆、预测、优化的通用机器智能方法，是人工智能

基础研究领域的一本优秀学术专著。作者何海波教授是人工智能领域内的一位杰出的青年学者，目前是美国罗德岛大学的讲席教授(Robert Haas Endowed Professor)、智能计算与自适应系统实验室主任，主要从事智能计算、控制与优化、机器学习、大数据、网络安全、大规模复杂系统等方向的研究。本书是作者及其团队近十年的研究积累，所述成果已产生深远影响，相信对我国从事人工智能研究的科技工作者大有裨益。

薛建儒

于西安交通大学

前　　言

目前理解类脑智能和研制有潜力再现自然智能的自适应系统仍是科学和工程领域尚未解决的最大挑战之一。随着人脑研究和现代技术不断取得新进展，科学家和工程师希望找到研制高度鲁棒、自适应、易扩展且可容错的通用类脑智能系统的正确途径。然而，要实现这一目标，还有很长的路要走。其中，最大的挑战是，如何理解智能的基本原理，开发有潜力捕获智能集成的复杂系统，最终使智力更接近真正智能。

本书的目的是促进理解和发展机器智能的自适应系统，并给出能够自适应地学习信息且随着时间积累知识、调节行为来实现目标的计算模型与体系架构。机器智能研究利用了许多学科的理论和概念，包括神经科学、人工智能、认知科学、计算理论、统计学、计算机科学、工程设计等。由于机器智能所固有的跨学科性质，所以这本书给出的大部分研究结果受不同领域的最新研究进展的启发。我希望本书的研究结果能够为理解机器智能的本质问题提供有用且重要的见解，并提供应用范围广泛的新技术和解决方案。

最近的研究结果证明，相比于传统的人工智能，类脑智能更加与众不同。比如，虽然如今的计算机可以解决非常复杂的数学问题，预测大规模的天气变化，甚至赢得世界象棋大赛，但是它们使用了与生物大脑有机体完全不同的信息处理方式。为此，这本书重点讨论机器智能的计算基础和方法论，目标是使自适应智能系统具备“计算思维”。所以，本书给出的研究结果可以自然地分为两类：**数据驱动方法**和**生物启发式方法。**

数据驱动方法的目标是理解如何设计自适应系统，使它能从大量的原始数据中自主学习信息和知识表达，以支持不确定和非结构化的环境中的决策过程；生物启发式方法的目标是理解在分布式分层神经网络内部信息处理的原则、关联、优化，以及预测。所有这些将来都会被用于实现通用类脑机器智能的基本功能和特性。在本书的最后一章，我对机器智能研究的硬件设计给出，如专用超大规模集成(VLSI)系统，以及可重构的现场可编程门阵列(FPGA)技术，这提供了如何用大规模、并行和可伸缩的硬件平台构建复杂且综合的智能系统的有用的建议。最后一章还简要地讨论了新兴技术(如忆阻器)，因为这些技术可能为我们提供重要的新功能以模拟

复杂的人类大脑神经结构。此外，为了突出机器智能研究的广泛应用，每章末尾都配有一个案例研究，以说明本书所提供的方法能有效应用于不同领域。这些例子为应用本书提到的方法提供了有用的建议。

本书分为4个主要部分，组织结构如下：

1. 第一部分(第1章)简要介绍机器智能自适应系统，给出了研究意义以及传统计算机与类脑智能的主要区别，简要说明了本书的组织结构，并介绍本书的使用方法。

2. 第二部分(第2～4章)介绍数据驱动的机器智能研究方法。重点是开发自适应学习方法，将大量的原始数据转换成知识和信息表达，从而支持不具确定性的决策过程。主要介绍增量学习、不平衡学习和集成学习。

3. 第三部分(第5～7章)重点讨论生物启发式机器智能研究。其目标是理解神经信息处理的基本原则，并开发学习、记忆、优化和预测架构的计算来模仿特定水平的智能。主要介绍自适应动态规划(ADP)、联想学习和序列学习。

4. 第四部分(第8章)简要讨论机器智能的硬件设计。其目标是提供设计硬件系统时要重点考虑的一些因素，例如：功耗、设计密度、内存需求和速度需求，目的是实现大规模、复杂的综合智能系统硬件。

本书面向机器智能领域学术界和工业界的研究人员，书中的学习原理、体系结构、算法和实例研究，不仅有望为机器智能研究领域带来新见解，而且提供了面向广泛应用的机器智能技术和解决方案。此外，书中讨论的所有问题都属于相关领域内具有重大挑战性的热门研究主题，这使得本书成为研究生激励自己向博士研究项目或大师级研究水平迈进的宝贵资源。最后，由于机器智能研究在不同的学科中不断地引起越来越多的关注，因此我也希望这本书能够提供有趣的观点和建议，以激发大学生和年轻研究者对这个激动人心且有价值的领域中的科学和技术问题产生浓厚的兴趣，他们的参与对这个健康且有前途的研究领域的长期发展至关重要。

致　谢

十分感谢许多同事、朋友、审稿人和学生，感谢他们在这一领域的研究以及对本书的写作提供的帮助！

非常感谢罗德岛大学(URI)和史蒂文斯理工学院(SIT)的同事与朋友们在本书写作过程中所提供的巨大支持。罗德岛大学电子、计算机、生物医学工程(ECBE)系和工程学院(COE)的许多同事以不同的形式为本书提供了巨大的支持，特别感谢 G. Faye Boudreaux-Bartels、Raymond Wright、Qing (Ken) Yang、Yan(Lindsay) Sun、He (Helen) Huang、Steven M. Kay、Godi Fischer、Leland B. Jackson、Walter G. Besio、Peter F. Swaszek、Frederick J. Vetter、Resit Sendag、Richard J. Vaccaro、Ying Sun、Harish Sunak、Ramdas Kumaresan、Jien-Chung Lo、William J. Ohley、Shmuel Mardix 和 Augustus K. Uht，他们对我在该领域的研究和教育提供了支持。史蒂文斯理工学院电子与计算机工程(ECE)系、舍费尔工程与科学学院的许多朋友和同事也为我的研究工作提供了巨大的支持。另外，还要特别感谢 Joseph Mitola III、Yu-Dong Yao、George Korfiatis、Michael Bruno、Stuart Tewksbury、Victor Lawrence、Yi Guo、Rajarathnam Chandramouli、Koduvayur Subbalakshmi、Harry Heffes、Hong Man、Hongbin Li、Jennifer Chen、Yan Meng、Cristina Comaniciu 和 Bruce McNair 对我在这个领域的研究工作所提供的巨大支持。

深深感谢我的学生们和这些年一起工作的访问学者，尤其要感谢 Sheng Chen、Yuan Cao、Bo Liu、Qiao Cai、Jin Xu、Jie Li、Jian Fu、Jianlong Qiu、Yi Cao、Zhen Ni、Hao Peng、Edwardo A. Garcia、Xiaochen Li 和 Yang Bai，感谢他们有价值的讨论、评审，以及对本书的审校工作。同时，还要感谢听我讲课的学生们，感谢他们针对本书内容的相关建议和讨论(尤其是听 ELE 594——计算智能与自适应系统和 CpE / EE 695——应用机器学习课程的学生)。虽然没有提及他们的名字，但没有他们的帮助，这本书是不可能完成的。

其他大学、研究实验室和工业合作伙伴的许多朋友们也为我的研究以及本书的写作提供了极大的支持。要特别感谢 Janusz A. Starzyk 一直以来的支持和帮助，本书展现的很多资料来自于与他的讨论和共同研究的启发。也非常感谢 Xiaoping Shen

在机器智能研究数学方面提供的巨大支持。此外，Venkataraman Swaminathan、Sachi Desai、Shafik Quoraishee、David Grasing、Paul Willson 以及美国陆军装备研究开发和工程中心(ARDEC)的许多其他成员，在过去的几年里也给我提供了很大的支持，包括实际应用案例研究、真实环境数据集以及几个研究项目的技术讨论。我还想借此机会感谢 Charles Clancy、Tim O'Shea、Ray Camisa 和 Jeffrey Spinnanger 在各种会议中对许多先进技术的讨论以及对我在这个领域的研发的大力支持。

还要感谢许多国际上的专家和科学家，他们花费了很多宝贵的时间审阅资料，并为本书提供建议。虽然没有提及所有的名字，但我特别感谢以下专家的大力支持：Derong Liu、Jennie Si、Jun Wang、Gary Yen、Robert Kozma、Donald C. Wunsch II、Danil Prokhorov、Marios M. Polycarpou、Mengchu Zhou、Shiejie Cheng、Ping Li、Yaochu Jin、Kang Li、Daniel W Repperger、Wen Yu、Anwar Walid、Tin Kam Ho、Zeng-Guang Hou、Fuchun Sun、Changyin Sun、Robi Polikar、Jinyu Wen、Tiejian Luo、Xin Xu、Shutao Li、Zhigang Zeng 等。他们提供的专业知识极大地帮助了我在这一领域的研究。

另外，也非常感谢美国国家科学基金会(NSF)、美国国防部高级研究计划局(DARPA)，以及陆军装备研发和工程中心(ARDEC)这些年来在研发方面对我的巨大支持。他们的巨大支持为我探索这一领域的所有挑战和令人兴奋的研究课题提供了机会。

John Wiley & Sons 为这本书的完成提供了很大的支持。借此机会，要特别感谢 George J. Telecki 和 Lucy Hitz 所提出的宝贵建议和鼓励。如果没有他们的帮助，这本书的写作和出版将会花费很多时间。

最后，我想对我的家人致以最深切的感谢，尤其是我的妻子 Yinjiao，感谢他们的大力支持。我还想把这本书送给我可爱的小家伙——Eric。

何海波

目　录

第1章
绪　论

1.1 机器智能研究

由于理解类脑智能和研制有潜力复制达到相当于大脑智能水平的自适应系统还是尚未解决的科学和工程最大挑战，大脑在不确定和非结构化的环境中表现出强大的学习、记忆、预测和优化能力，为达成此目标提供了很强的证据。尽管神经科学研究在理解大脑智能基本机制方面取得了非常重要的进展，而且最新的技术发展使得构建复杂智能系统成为可能，但仍然不清楚该如何设计真正意义上通用的、能复现的智能机器（Werbos，2004，2009；Brooks，1991；Hawkins & Blakeslee，2004，2007；Grossberg，1988；Sutton & Barto，1998）。实现这一长期目标对科学和工程研究的多个学科提出了挑战，包括但不限于以下领域：

- 理解生物脑组织中神经信息处理的基本原理和机制。
- 发展通用机器智能学习、记忆、预测和优化的原则性方法。
- 研制能将大量原始数据转换为知识与信息表示的适应性模型和计算架构，以支持不确定的决策过程。
- 设计能通过与环境交互的学习并具有目标导向行为的智能硬件系统。
- 设计鲁棒、可扩展和可容错的系统，为复杂、集成化和网络化系统提供大规模并行处理硬件。

为了解决这些挑战性问题，许多学科都致力于这一领域的研究，包括神经系统科学、人工智能、认知科学、计算理论、统计学、计算机科学和工程设计等。例如，人工神经网络在模拟类脑学习功能的建模中发挥着重要作用(Grossberg，1988)。反向传播理论为构建智能系统提供了一个强有力的方法，并在包括模式识别、自适应控制和建模、灵敏度分析等(Werbos，1988a，1988b，1990，2005)领域取得成功。在这个领域还有许多其他代表性的工作，包括记忆预测理论（Hawkins & Blakeslee，2004，2007)、强化学习(RL)(Sutton & Barto，1998)，具身智能(Brooks，1991，

2002)、自适应动态规划(ADP)(Werbos，1997，1998，2004，2009；Si，Barto Powell & Wunsch，2004；Powell，2007)、“新人工智能”理论(Pfeifer & Scheier，1999)等。例如，最近，为了设计智能机器，提出了基于分层记忆组织的新理论框架(Hawkins & Blakeslee，2004，2007)。这种理论框架为如何理解皮层神经的记忆与预测机制提供了新的有潜力的解决方案。由于生物智能系统可以通过与外部环境的积极互动进行学习，因此，强化学习在业界备受关注，并在多个领域(Sutton & Barto，1998)有成功应用。强化学习的核心思想是学习如何建立情景到动作的映射，使得期望的奖励信号最大。价值函数是强化学习的本质特性之一，它通过指定“好”与“坏”来指导智能系统的目标导向行为。例如，在生物系统中，它可能是一种测量快乐或痛苦的方法(Starzyk，Liu & He，2006)。具身智能的思想源于对具有生物机体、能适应一定的真实环境的生物智能的观察(Brooks，1991，2002)。具身智能的研究重点集中在理解生物智能、发现智能行为的基本原理及设计实际智能系统，包括活体机器和人形机器人。最近，我们认识到，优化和预测在使类脑通用智能更接近真实的过程中发挥着至关重要的作用(Werbos，2009)。例如，最近美国国家科学基金会启动的认知优化和预测(COPN)计划表现出了对这一关键领域的关注。该计划组织跨学科团队合作解决大脑如何学习解决复杂优化和容错控制的根本问题(NSF，2007)。虽然优化在控制理论、决策理论、风险分析和许多其他领域有着长期的研究基础，但在机器智能研究方面有特定意义：优化是指通过长时间学习，做出更好的选择，从而最大化实现目标的某些效用函数。大量研究工作表明 ADP 是核心方法，或者是“在通常情况下学习逼近最优行动策略的唯一通用途径”(Werbos，2004，2009)。当然，应该指出的是，上述提到的许多领域具有很强的关联关系。例如，ADP / RL 方法可用“具身”(与感知运动相结合，协调与外部环境的积极交互)或分层方式构建有效的面向目标的多级学习、预测和优化(Werbos，2009)。

从实际应用的角度来看，新技术的发展使获取的原始数据以爆炸性速度产生，例如，传感器网络、安全和防御应用、互联网、地理信息系统、交通运输系统、天气预报、生物医药产业、金融工程等。上述诸多应用面临的挑战不是缺乏原始数据，而是信息处理速度、与原始数据爆炸性增长速度及将其转化为可用形式的速度无法相适应。这就为机器智能行业带来了巨大的机会和挑战：开发自适应系统来处理巨量原始数据，以支持决策过程中的信息表示与知识积累。

因此，本书重点介绍设计具有“计算思维”(Wing，2006)的自适应智能系统的机

器智能研究的计算基础。例如，尽管传统的人工智能方法已经在不同的具体应用任务中取得了显著的进步和巨大的成功，但这些技术不具备可用于不同知识领域的鲁棒性、可扩展性和适应性。但是，生物智能系统能通过自适应地学习，不断积累知识，从而实现任务导向行为。例如，尽管当今的计算机可以解决非常复杂的问题，但它们使用了与人脑根本不同的信息处理方法(Hawhins & blakeslee，2004，2007；Sutton & Barton，1998)。这就是为什么一个3岁大的婴儿可以轻松地观看、聆听、学习和记忆各种外部环境信息，并相应地调整他或她的行为，但最尖端的计算机却不能如此。这就提出了一些关键问题，如“人类能把什么做得比计算机更好或反之亦然?”，或者，更重要的，从计算思维的角度来看“什么是可计算的?”(Wing，2006)。我们相信，相关这个问题的深入理解对机器智能研究是至关重要的，并能最终提供实用技术和解决方案，有望在不同领域实现更接近现实的智能水平。

为了简要概述传统计算与类脑智能之间的主要区别，图1-1比较了这两种不同智能水平的主要特征。我们可以清楚地看到，对所有的关键任务，类脑智能与传统计算有着本质区别。因此，从计算思维的角度看，研制类脑智能需要新的认知、基础、原则和方法。本书力图提供这一领域的最新研究进展，以满足上述需求。

传统计算	任务	类脑智能
顺序的	信息处理	并行的
固定的	复杂性	可伸缩的
集中式的	控制机制	分布式的
全局的	交互作用	局部的
程序化	行为来源	自组织/概念化
有限的、受限的	容错性	高
自定义设计	架构	进化的
一些	适应性	高
特定应用	应用领域	鲁棒的

图1-1 传统计算与类脑智能的比较

1.2 两个层面：数据驱动方法和生物启发式方法

图1-2展示了本书重点讨论的机器智能框架的高层视图。其中，有两个重要组成部分：智能核，如神经网络组织和学习规则智能核与外部环境通过感觉

运动通道(具身)的交互。因此，本书包括两个部分，从两个主要层面阐述机器智能：机器智能研究的数据驱动方法和生物启发式方法。这样，我们不但能了解神经网络组织的基础和原理，以及智能核的学习方法，而且还能促使我们通过关注数据处理途径(感知、采集、处理和行动)来改进主要方法。这里的关键问题包括：类脑系统与非结构化和不确定环境如何自适应交互、如何处理大量原始数据、如何发展内部结构、如何建立关联和预测等、如何随着时间推移积累知识并最终利用自我控制实现目标。

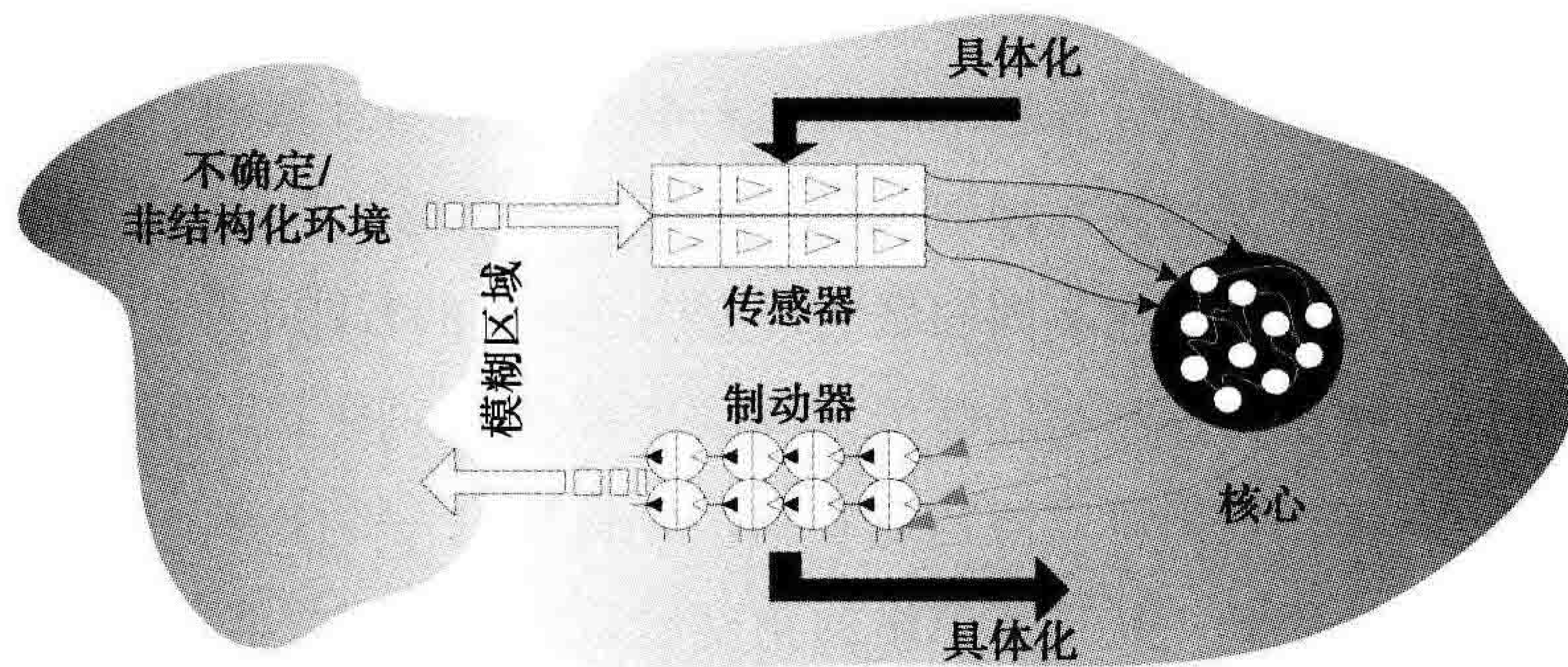


图 1-2　机器智能的高级视图

数据驱动方法的根本动机非常直接：数据是任何种类的信息处理、知识转化和决策过程的原始资源。从计算智能的观点来看，数据几乎涉及“智能”的方方面面：推理、规划和思考等。因此，不同形式的数据对机器智能的发展起着至关重要的作用，如感知、采集、处理、转化和利用。从这个角度看，可以联想到从办公桌上拿起一支笔、开车行驶在纽约的繁华街区、安排下个月的日程表等许多实际例子。所有的这些任务都涉及不同层次的数据分析。若要设计一个最大可能复制特定层次水平的类脑智能的智能机器，从数据计算的角度出发，会遇到许多核心问题，如：什么类型的数据在决策过程中是必要的？一个智能机器如何能够不断地从不稳定和有噪声的数据中学习？如何有效地基于不同数据空间的不同假设并结合多重投票得到最优决策？

具体而言，本书将讨论下述机器智能研究的数据驱动方法：

增量学习：增量学习对于理解类脑智能是至关重要的，并至少在两个方面有潜力使机器智能更接近真实智能：①智能系统在整个生命周期内应能持续地增量式学习与准确体验，并利用这些积累的知识促进未来的学习和决策过程；②智能系统与环境交互产生的原始数据在无限期(可能是无限)学习生命周期内不断递增。这些学习情境完全不同于传统的静态学习，因为在静态学习中，一个有代表性的数据分布

用于训练，训练数据表示得到决策边界，用决策边界对未来的数据做出预测。进一步，如何通过增量学习实现全局性泛化是正确理解这些问题的核心部分。因此，用超越传统的“计算一存储一检索”方法开发自然智能系统，对于形成大规模复杂数据处理系统非常重要。

不平衡学习：许多实际应用要求智能系统从发生形变的数据分布中学习，支持决策过程。这些形变的数据分布中未被充分表示的数据会显著影响系统的学习能力和性能。例如，现存的许多学习算法假设或期望利用平衡数据分布来确定决策边界。因此，若碰到不平衡数据，这些学习算法就不能正确表示数据的分布特征，最终导致学习性能变差。由于不平衡数据固有的复杂特征并且频繁地出现在许多实际系统中，因此不平衡学习问题已经成为许多应用领域和尚未涉及的领域中的新的挑战性问题。

集成学习：一般来说，与单模型学习方法相比，集成学习方法具有改进精确性和鲁棒性的优点。集成学习就是使用多个分类器，通过投票方法进行决策组合来实现预测。由于不同的分类器可以提供不同的目标函数，相对于单模型学习方法，组合决策有望提供更鲁棒和更准确的决策。与集成学习相关的重要因素有两个：①给定训练数据后，如何设计多分类器？为了得到多分类器，可利用如自举聚集(bagging)、自适应增强、随机子空间、层叠泛化、混合专家系统等多种方法。②如何有效地整合多个分类器的输出，以获得比单个分类器更好的决策？这主要包括不同类型的组合投票策略(也将在本书中讨论)。

除了数据驱动方法，本书还介绍了开发机器智能的生物启发式方法。最新的脑科学研究提供的证据表明，与今天的计算机对比，生物大脑使用完全不同的方式处理各种任务(Hawkins & Blakeslee，2004，2007；Hedberg，2007)。例如，10多年前，IBM的Deep-Blue在国际象棋比赛中能赢得世界冠军，但这并不能告诉我们可以用完全与人脑处理信息不同的方式开发出通用的类脑智能机器。另一方面，在大师级象棋程序的自学习能力开发方面，进化算法表现出巨大的潜能，这让我们能更好地理解机器智能的本质(Fogel，Hays，Han & Quon，2004)。从这个角度来看，关键是如何开发能够模拟相当大脑智能水平的生物启发式系统模型和体系结构。在本书中，我们将讨论关于这个问题的三个主要部分。

(1)自适应动态规划(ADP)

ADP已经被广泛认为是理解和再现通用类脑智能的关键方法(Werbos，1994，1997，2004，2009；Si等，2004；Powell，2007)。为了促进机器智能研究，ADP

研究的两个主要目标分别是**优化**和**预测**。特别地，优化可定义为随着不断学习能做出使效用函数最大化的更好的选择，最终实现目标(Werbos，2009)。为此，在随机系统中，优化的基础是 Bellman 方程(Bellman，1957)与 Von Neumann 提出的基数效用函数紧密结合。除了优化，最新神经生物学研究的证据表明，预测是提供一定水平的自适应通用智能的又一重要因素(Werbos，2009)。在 ADP 设计中，预测可以被认为是更一般的方法，包括许多重要信息，如来自观察数据的未来感官输入，以及对未观察到的状态变量的建模和重构，具有优化选择动作的目的(Werbos，2009)。本书提出了一种具有多目标表示的分层 ADP 结构，可以更有效地整合优化和预测，从而进行机器智能研究。

(2)联想学习

联想记忆在基于信息联想与预测的自然智能中起着重要作用。一般来说，有两种类型的联想记忆：异联想记忆和自联想记忆。异联想记忆能够把匹配的模式关联起来，例如文字和图片；而自联想记忆把一种模式与自身相联系，并从部分模式中通过联想回忆出不完整的部分，恢复完整的模式。人脑同时使用异联想和自联想记忆来学习、规划行为和预测(Rizzuto & Kahana，2001；Brown，Dalloz & Hulme，1995；Murdock，1997)。大脑的记忆是自组织的和数据驱动的。自组织不仅在人脑中负责分层组织结构的形成，而且在低等脊椎动物的神经系统中也负责分层组织结构的形成(Malsburg，1995)。本书将聚焦于联想学习的本质特征研究，包括自组织、稀疏和局部连接、分层结构。

(3)序列学习

序列学习被广泛认为是人类智能中最重要的组成部分之一，因为大多数的人类行为是一系列的行为模式，包括但不限于自然语言处理、语音识别、推理和计划等。因此，理解序列学习的基本问题对机器智能的研究具有至关重要的作用。为此，针对分布式分层神经组织内的序列学习、存储和检索，本书提出了一种生物启发模型。预测能力是这个序列学习模型中的关键要素。

1.3 如何阅读本书

本书包括机器智能背景的介绍性讨论，以及该领域最前沿的理论和实践发展。本书不仅提供了数学基础、学习原理、模型和算法，还提供了大量的实例研究，以说明这些方法在解决实际问题中的应用。所有的这些都为对机器智能领域感兴趣的研究人员和从业人员提供了有价值的资源。

本书也可以作为机器学习、数据挖掘、计算智能、自适应系统领域的研究生和本科生教材。对于重点在数据处理方面的课程，建议学生学习本书的第2～4章。对于偏重于生物启发学习的课程，学生可以学习第5～7章。第8章提供了对于大规模并行处理体系结构的机器智能硬件设计的有趣讨论，既包括专用的超大规模集成电路系统，又包括可重新配置的现场可编程门列阵技术，还介绍了现有新兴技术(包括忆阻器)是如何通过在硬件中建立复杂的智能系统从而改变我们的社会的。本书的所有内容可作为一个学期的机器智能课程。此外，为了适应不同程度/级别教学的要求，本书各个章节的内容安排得尽可能相对独立，以便学生可按不同顺序阅读，但出于内容上的系统性和一致性的考虑，部分章节之间的相关性也无可避免。

图1-3显示了本书的组织结构，各章节的简要综述总结如下。

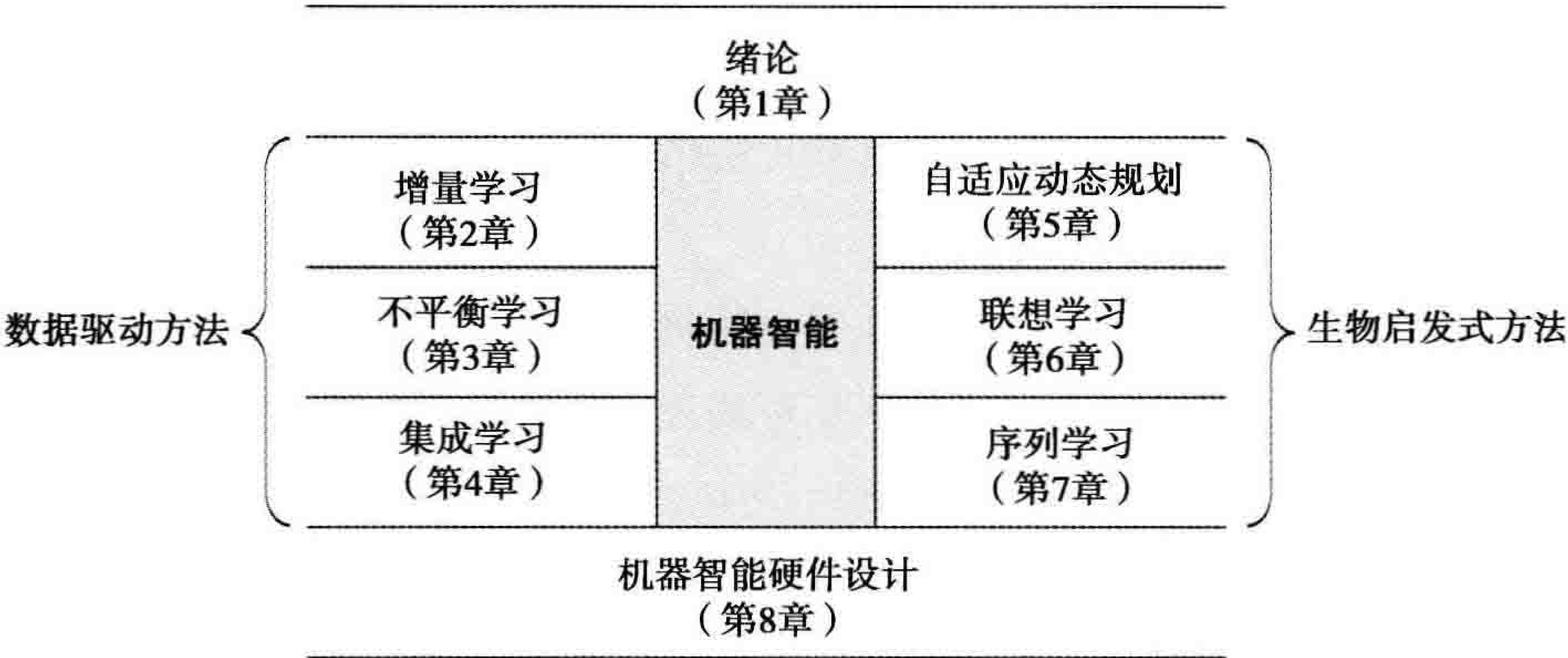


图1-3 本书的组织结构

1.3.1 机器智能的数据驱动方法

第2章介绍了增量学习的基本概念及其对机器智能研究的重要性。重点是理解随着时间积累知识以支持决策过程中的自适应学习原则和基础。本章给出了具体的学习框架和各种实际设计中应该考虑的因素。用视频数据流和电子邮件数据的应用研究实例证明了这个方法的有效性。

第3章介绍不平衡学习问题。作为该领域中相对新的方向，本章的重点是了解不平衡学习问题的本质，并且运用当前最先进的技术来解决这个问题。方法论中提到的4类方法包括：抽样法、代价敏感法、基于核的学习方法和主动学习方法。本章也讨论了不平衡学习的性能评价指标及不平衡学习的主要挑战

和发展机遇。

第 4 章介绍集成学习方法。本章的重点是设计多样性学习分类器，并整合这些多分类器来支持最终决策过程。其中，讨论的主要方法包括自举聚焦、自适应增强(AdaBoost)和子空间法等。本章还涉及许多组合投票策略和详细的间隔分析。

1.3.2 机器智能的生物启发式方法

第 5 章讨论了机器智能研究中的 ADP 方法，重点是了解优化和预测的 ADP 基本设计原理，提出了 3 种网络类型的分层学习架构和基于反向传播的具体学习算法，给出了这种架构在倒立摆平衡控制问题研究中的一个实例。

第 6 章介绍了分层神经网络组织中的自组织记忆，主要包括联想学习机制、神经网络组织和网络操作，以及记忆组织的应用实例：异联想和自联想。

第 7 章介绍了复杂序列学习、存储和检索的神经网络结构。这种用于机器智能的架构具有两个重要特征：分层神经组织和分布式信息处理。我们提出了详细的系统级架构及其学习机制。最后用一个具有 4 层结构的实例说明了该模型在文本分析中的应用。

最后，第 8 章讨论了关于最新硬件平台的机器智能硬件设计，包括专用的超大规模集成电路技术，以及可重配置的现场可编程门列阵技术。本章的目标是为在硬件设计时需考虑的因素提出建议，例如：功耗、设计密度、内存需求和速度需求，目的是实现大规模、综合且复杂的智能系统硬件。作为本书的结束语，还介绍了许多新兴技术，例如忆阻器，以用于未来类人脑智能系统的设计。

1.4 总结和延伸阅读

本书旨在提高读者对通用类脑智能研究的基本理解，发展原则性的方法和实用技术，并通过广泛应用，产生在一定程度上更加接近现实的机器智能。本书介绍的基本方法是基于机器智能研究的两条路径：数据驱动方法和生物启发式方法。

机器智能研究借鉴了许多学科的理论和概念。对于这一领域最新的研究发展，有兴趣的读者可以从许多国际期刊上找到大量好的资源，具体包括，当然也不仅限于这些：*IEEE Transactions on Neural Networks*，*Neural Networks*，*Neural Computation*，*IEEE Transactions on Evolutionary Computation*，*IEEE Transactions on Pattern Analysis and Machine Intelligence*，*IEEE Transactions on Knowledge and Data Engineering*，*Artificial Intelligence*，*Cognitive Brain Research*，*Machine Learning* 等。同时，也有许

多学术会议涵盖了机器智能研究的不同方面，包括 *International Joint Conference on Neural Networks*（*IJCNN*），*National Conference on Artificial Intelligence*（*AAAI*），*International Joint Conference on Artificial Intelligence*（*IJCAI*），*Neural Information Processing Systems*（*NIPS*），*International Conference on Machine Learning*（*ICML*），*International Conference on Data Mining*（*ICDM*），*Annual Meeting of the Cognitive Science Society*（*CogSci*）等。

参考文献

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Brooks, R. A. (1991). Intelligent without reason. *Proc. Int. Joint Conf. on Artificial Intelligence*, pp. 569–595.

Brooks, R. A. (2002). *Flesh and machines: how robots will change us*. New York: Pantheon.

Brown, G. D. A., Dalloz, P., & Hulme, C. (1995). Mathematical and connectionist models of human memory: a comparison. *Memory*, *3*(2), 113–145.

Fogel, D. B., Hays, T. J., Han, S. L., & Quon, J. (2004). A self-learning evolutionary chess program. *Proc. IEEE*, 92, 1947–1954.

Grossberg, S. (1988). *Neural networks and natural intelligence*. Cambridge, MA: MIT Press.

Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: Times Books.

Hawkins, J., & Blakeslee, S. (2007). Why can't a computer be more like a brain? *IEEE Spectrum*, *44*(4), 20–26.

Hedberg, S. R. (2007). Bridging the gap between neuroscience and AI. *IEEE Intel. Syst.*, *22*(3), 4–7.

Malsburg, C. V. (1995). Self-organization and the brain. In M. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks* (pp. 840–843). Cambridge, MA: MIT Press.

Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (todam2). *Psychological Review*, *104*, 839–862.

NSF. (2007). Emerging frontiers in research and innovation: Cognitive optimization and prediction: From neural systems to neurotechnology (copn). *[online], available: http://www.nsf.gov/pubs/2007/nsf07579/nsf07579.htm.*

Pfeifer, R., & Scheier, C. (1999). *Understanding intelligence*. Cambridge, MA: MIT Press.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*. Hoboken, NJ: Wiley.

Rizzuto, D. S., & Kahana, M. J. (2001). An autoassociative neural network model of paired-associate learning. *Neural Computation*, *13*, 2075–2092.

Si, J., Barto, A., Powell, W. B., & Wunsch, D. (2004). *Handbook of learning and approximate dynamic programming*. Piscataway, NJ: IEEE Press.

Starzyk, J. A., Liu, Y., & He, H. (2006). Challenges of embodied intelligence. *Proc. Int. Conf. on Signals and Electronic Systems*. Lodz, Poland.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Werbos, P. J. (1988a). Backpropagation: Past and future. *Proc. IEEE Int. Conf. Neural Netw.*, I-343–353.

Werbos, P. J. (1988b). Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.*, *1*, 339–356.

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proc. IEEE*, 78, 1550–1560.

Werbos, P. J. (1994). Approximate dynamic programming for real time control and neural modeling. In P. J. White & P. J. Sofge (Eds.), *Handbook of intelligent control* (pp. 493–525). New York: Van Nostrand.

Werbos, P. J. (1997). Brain-like design to learn optimal decision strategies in complex environments. *Proc. Conf. Decision and Control*, pp. 3902–3904.

Werbos, P. J. (1998). A brain-like design to learn optimal decision strategies in complex environments. In M. Karny, K. Warwick, & V. Kurkova (Eds.), *Dealing with complexity: A neural networks approach*. London: Springer.

Werbos, P. J. (2004). ADP: Goals, opportunities and principles. In J. Si, A. G. Barto, W. B. Powell, & D. Wunsch II (Eds.), *Handbook of learning and approximate dynamic programming* (pp. 3–44). Piscataway, NJ: IEEE Press.

Werbos, P. J. (2005). Backwards differentiation in AD and neural nets: Past links and new opportunities. In H. M. Bucker, G. Corliss, P. Hovland, U. Naumann, & B. Norris (Eds.), *Automatic differentiation: Applications, theory and implementations, lecture notes in computational science and engineering* (Vol. 50, pp. 15–34). Berlin, Germany: Springer-Verlag.

Werbos, P. J. (2009). Intelligence in the brain: A theory of how it works and how to build it. *Neural Networks*, *22*, 200–212.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

第2章 增量学习

2.1 引言

增量学习是一种重要的类脑智能功能，因为生物系统能终生持续不断地学习并随时间积累知识。从计算的角度看，研究机器智能的增量学习有3个关键目标：把以前学到的知识转化为当前接收到的数据，以促进对新数据的学习；不断随时间积累经验，以支持决策过程；通过学习实现全局泛化，以达成目标。这不仅是开发主要学习方法以了解类脑系统如何自适应地处理和学习原始数据的需要，而且对许多涉及巨量流数据的实际应用同样重要。在增量学习时，智能系统与环境交互产生的数据在无限期的学习周期中不断递增。这种学习完全不同于传统的静态学习任务，因为在静态学习中，训练数据所表示的分布有代表性有效的，可用来确定决策边界。本章提出一种自适应增量学习框架来解决这个问题。

2.2 问题的提出

考虑如下情况：用 D_{j-1} 表示在 $t_{j-1}\sim t_j$ 期间接收到的数据块，分类假设 h_{j-1} 是基于 D_{j-1} 的。重要的问题是：当接收到一个新的数据块 D_j 时，系统该如何自适应地学习？针对这个问题有两类主要的传统方法。

第一类是使用如图2-1a所示的简单数据累积方法。在这类方法中，当接收到新数据块 D_j 时，丢弃 h_{j-1}（用“×”表示），并利用到当前为止所有的累积数据（…，D_{j-1}；D_j）来设计新分类器 h_j。这是一个非常直观的方法，它没有考虑利用在 h_{j-1} 中已经学到的知识或者经验来帮助从新数据的学习。此外，由于内存和计算资源有限，大多的实际应用无法存储所有累积的数据。

第二类是如图2-1b所示的集成学习方法。简单地讲，当新数据块可用时，基于新数据设计单个分类假设 h_j 或者分类假设集合 H：h_i，$i=1,\cdots,L$。然后，用投票

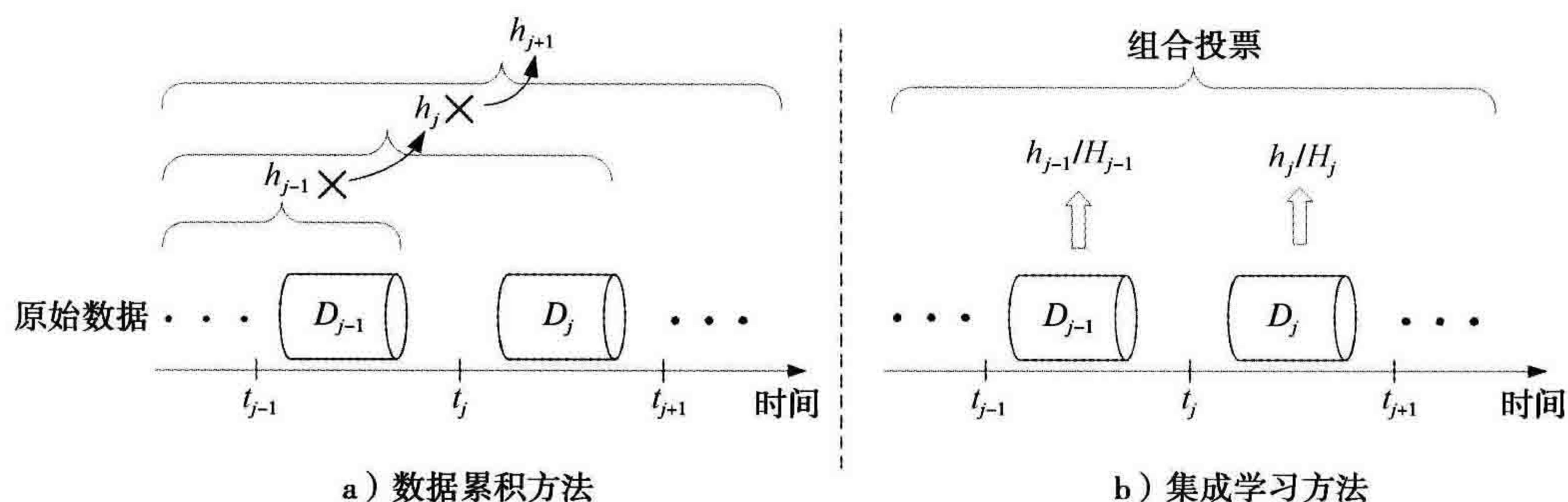


图 2-1　增量学习的两类传统方法

机制融合所有不同分类假设的决策以得到最终预测。这类方法的主要优点是无须存储或访问以前的观测数据，因为通过学习得到的一系列分类假设中已含有了这些知识。然而，这种方法认为每个数据块是独立的知识表示，因此不存在从旧知识到新知识的经验累积和知识转化。例如，在$[t_{j-1}, t_j]$时间段学到的知识不能用于$[t_j, t_{j+1}]$时间段的学习，尽管这两个时间段的分类假设都会参加最终投票。此外，学习获得的一系列分类假设 h(或分类假设集合 H)仍然是一个域模型：无法保证这样的学习方法能产生可用于整个数据流(或至少其中的一部分)的结果。唯一的知识集成过程出现在最后的投票阶段。因此，这类方法忽略了机器智能中增量学习的本质问题，即随时间不断累积经验且将来的学习能从中受益。

2.3　自适应增量学习框架

AdaBoost 方法已经得到广泛研究，并在机器学习和数据挖掘研究中显示出极大成功(Freund & Schapire，1996，1997；Freund，2001；Schapire，Freund，Barlett & Lee，1998；Oza，2003，2004；Dietterich，2000；Bauer & Kohavi，1999)。理论上已经证明：弱学习器的最终分类假设的训练误差以指数级速度下降到零。AdaBoost 算法对错误分类的样本(“困难”的样本)赋予相对于能正确分类的样本(“容易”的样本)更高的权重(Freund & Schapire，1996，1997)。因此，基于分类假设评估，样本权重不断被迭代更新，决策边界自动地更加关注困难的样本。对连续数据进行增量学习的关键问题是如何根据以前的知识对接收到的新数据更新权重。如 2.2 节的讨论，现有的许多基于自举思想的增量学习方法针对新训练数据的不同子集创建多个分类假设，然后通过集成学习来实现最终分类器。这种方式没有涉及通过整个学习过程累积知识用于下一步学习和决策过程的关键部分。对于自适应智能系统，这并非是使用自适应自举的本质思想获得所期望的增量学习能力的自然方式。

假设一个智能系统可表示为随时间变化的数据流，在时刻 t 收到一个新训练数据集 D_t。这种情况下以前的知识为在时刻 $t-1$ 由数据集 D_{t-1} 的分布函数 P_{t-1} 得到的分类器 h_{t-1}。这里的分布函数为抽样概率函数或权重分布函数：难于学习的困难样本的权重高于易于学习的样本(Freund & Schapire，1996；1997)。这里的目标有3个方面：由以前的数据到当前数据的知识转化，随时间的经验累积和达到全局泛化。系统级框图如2.2所示，下面描述详细的学习算法。

为了实现上述目标，这个框架含有3层组织和3个数据流方向。第1层(D 层)为学习系统的输入、原始数据流，这里用$(x_i,y_i)(i=1,\cdots,m)$表示数据集 D_t，其中 x_i 表示 n 维特征空间 X 的一个样本，$y_i\in Y=\{1,\cdots,C\}$表示 x_i 对应的类别标签。第2层(P 层)是反映数据可学习性的分布函数层，基于先前累积的经验将原始数据转化为知识表示。第3层(H 层)通过将数据与 P 层整合以设计出支持最终决策过程的多个分类假设。通过这种方式，该框架间接地考虑将所有以前的域数据和累积的知识(没有“灾难性遗忘”(Grossberg，1998，2003))用于将来的任一时间段而无须访问以前的观测数据(如图2-2中的虚线箭头所示)。这与图2-1所示的传统方法有根本的不同。传统方法或需要明确地累积和存储原始数据(见图2-1a)，或需要假设每个数据块与知识累积不相关(见图2-1b)。

[算法2.1] 自适应增量学习

***t* 时刻的知识：**

- 数据集 D_t 有 m 个样本：$(x_i,y_i)(i=1,\cdots,m)$，其中 x_i 是 n 维特征空间 X 的一个样本，$y_i\in Y=\{1,\cdots,C\}$是 x_i 的类别标签。
- 分布函数 P_t，其中 $P_t=[w_1^t,w_2^t,\cdots,w_m^t]$，$\sum\limits_i P_t=1$。
- 基于数据 D_t 及其分布函数 P_t 的分类假设 h_t。

***t*+1 时刻的新输入：**

- 新数据集 D_{t+1}，可表示为$(x_i,y_i)(i=1,\cdots,m')$，m' 与 m 的大小可能相同，也可能不同。

学习过程：

1)寻求 D_t 与 D_{t+1} 的关系：

$$Q_t=\psi(D_t,D_{t+1}) \tag{2-1}$$

其中，ψ 是预先定义的映射函数，$Q_t=[\alpha_1^t,\alpha_2^t,\cdots,\alpha_{m'}^t]$为反映 D_t 与 D_{t+1} 关系的定量度量。

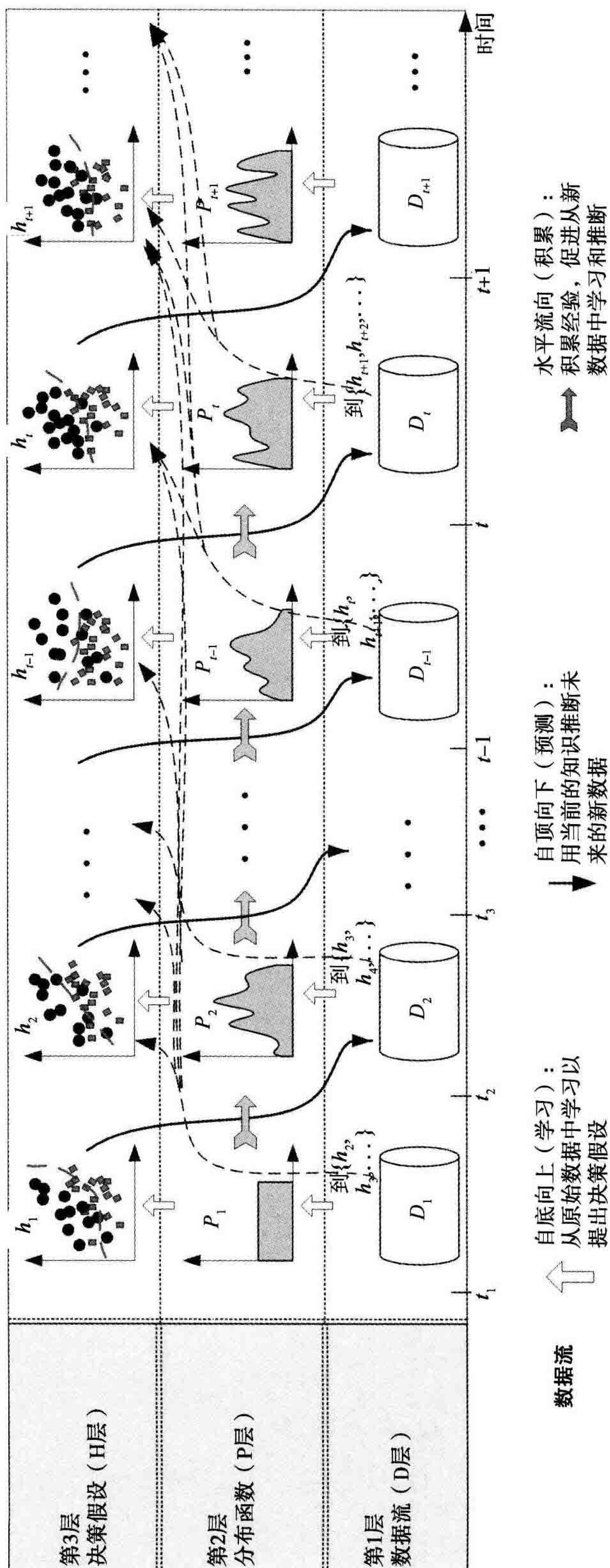


图2-2 自适应增量学习框架

2)更新 D_{t+1} 的初始分布函数：

$$\hat{P}_t = P_t \times Q_t \tag{2-2}$$

3)对 D_{t+1} 应用分类假设 h_t，计算 h_t 的伪误差：

$$\varepsilon_t = \sum_{j:h_t(x_j)\neq y_j} \hat{P}_t(j) \tag{2-3}$$

4)设 $\beta_t=\varepsilon_t/(1-\varepsilon_t)$。

5)改进 D_{t+1} 的分布函数：

$$P_{t+1} = \frac{\hat{P}_t}{Z_t} \times \begin{cases} \beta_t, & h_t(x_j) = y_j \\ 1, & 其他 \end{cases} \tag{2-4}$$

其中，Z_t 是归一化常数，P_{t+1} 是分布函数（$\sum_j P_{t+1} = 1$），可以表示为 $P_{t+1} = [w_1^{t+1}, w_2^{t+1}, \cdots, w_{m'}^{t+1}]$。

6)基于数据集 D_{t+1} 及分布函数 P_{t+1} 得到分类假设 h_{t+1}。

7)当接收新的数据集时，重复上述步骤。

输出：

最终的分类假设如下

$$h_{\text{final}}(x) = \arg\max_{y\in Y} \sum_{T:h_T(x)=y} \lg\left(\frac{1}{\beta_T}\right) \tag{2-5}$$

这里的 T 是在学习中以增量方式得到的分类假设集合。

自适应增量学习过程可以通过如图 2-2 所示的 3 个数据流方向完成。自底向上的数据流把原始数据转化为信息和知识表示：P_t 和 h_t。这里，P_t 是原始数据样本 D_t 的分布函数。那些难以学习的困难样本的权重高于容易学习的简单样本(Freund & Schapire，1996，1997)。基于 P_t 得到分类假设 h_t，对应的决策边界将会自动地更加关注比较困难的区域。对于第一个原始数据集 D_1，因为还没有开始学习，其分布函数 P_1 可以被设定为均匀分布。

获得 P_t 和 h_t 后，系统将使用这些知识学习原始数据块 D_{t+1}，如图 2-2 所示，这个过程是通过自顶向下和水平方向的数据流实现的。首先，当接收到新数据块 D_{t+1} 时，使用映射函数 ψ 建立 D_{t+1} 与 D_t 之间的分布关系，然后根据式(2-1)和式(2-2)所示的映射函数计算 P_{t+1} 的初始估计，记为 $\hat{P}_t$。用 $\hat{P}_t$ 的信息去评估 h_t 的学习能力，然后将它应用于新的数据块 D_{t+1}，并用式(2-3)计算累积误差。类似于 AdaBoost 算法(Freund & Schapire，1997)，β_t 是 ε_t 的函数。一般来说，β_t 代表了用以前的知识 h_t 学习新数据块 D_{t+1} 的优良性，学习算法通过数据流自适应地将决策

边缘推向那些困难的样本。一旦 P_{t+1} 由式(2-4)得到，就能设计新分类假设 h_{t+1}，在下一数据集中重复整个学习过程。最后，如式(2-5)所示，使用投票方法整合所有在增量学习过程中获得的知识，用于后续的决策过程。

总之，在这个增量学习框架中，水平方向的数据流使得系统随着时间积累经验和知识，有助于后续的学习和预测；而自顶向下方向的数据流能使系统运用积累的知识来预测接收到的数据流，然后完善学习能力。需要指出的是，这是一个通用增量学习框架，许多不同的基本学习方法(如决策树、神经网络等)均可以整合到这个框架中，可灵活用于不同应用领域的机器智能研究。

2.4 映射函数设计

增量学习框架(见 2.3 节)的一个关键是如何设计映射函数 ψ(见式(2-1))。函数 ψ 的目的是提供不同数据集分布函数的定量表示。使用自举思想解决静态学习问题的传统做法是对静态训练数据的分布函数以序贯的形式迭代更新(Freund & Schapire，1996，1997)。然而，增量学习时，新数据集的分布函数不能直接获得或更新。本书针对这种情况提出了 3 种解决方案：欧氏距离方法、回归学习方法和在线评估系统方法。

2.4.1 基于欧氏距离的映射函数

如图 2-3 所示，该方法用尺度欧氏距离寻求数据集 D_{t+1} 与 D_t 之间的关系，并估计初始分布 $\hat{P}_t$。

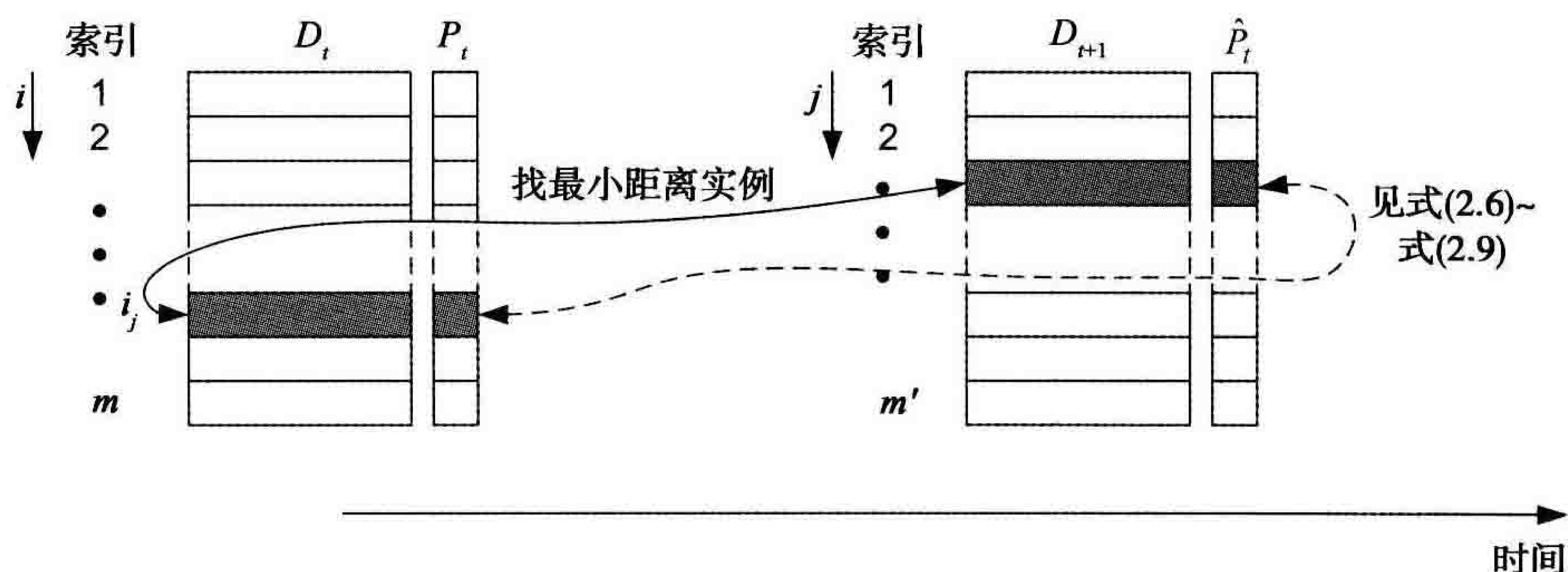


图 2-3 基于欧氏距离的映射函数

首先计算(D_{t+1}，D_t)的距离映射(DM)函数：

$$\mathrm{DM}_{ji} = \sqrt{\sum_{k=1}^{n} (x_{jk} - x_{ik})^2}, \quad j = 1,\cdots,m', i = 1,\cdots,m \tag{2-6}$$

$$I_j = \arg \min_{i \in \{1,\cdots,m\}} (\mathrm{DM}_{ji}) \tag{2-7}$$

$$\hat{Q}_j = \min(\mathrm{DM}_{ji}) \tag{2-8}$$

$I=[I_j]\in\{1,\cdots,m\}$是 D_{t+1} 中每个数据样本在 D_t 中的最近邻索引，$\hat{Q}=[\hat{Q}_j]\in[0,\infty)$是对应的距离值。距离 $\hat{Q}_j(j=1,\cdots,m')$确定后，对其进行缩放，即

$$Q_j = \frac{2}{1+\exp(\hat{Q}_j)} \tag{2-9}$$

这里，$Q_j\in(0,1]$。一旦 Q_j 确定，就可以用式(2-2)～式(2.4)更新 D_{t+1}的分布函数 P_{t+1}。

该方法如图 2-3 所示，可以看出，使用欧氏距离映射函数的主要思想是提供一个类似于聚类方法的连接关系，以便把以前的知识传递给新数据：数据集 D_{t+1}中任一数据样本 x_j，其初始分布 $\hat{P}_t(j)$可利用其数据集 D_t 中欧氏距离最小的样本所对应的 P_t 进行估计。

2.4.2 基于回归学习模型的映射函数

回归学习模型，如神经网络、支持向量机和决策树等，也可以集成进增量学习框架中来设计映射函数 ψ。图 2-4 所示为以神经网络模型为例的映射函数设计过程。

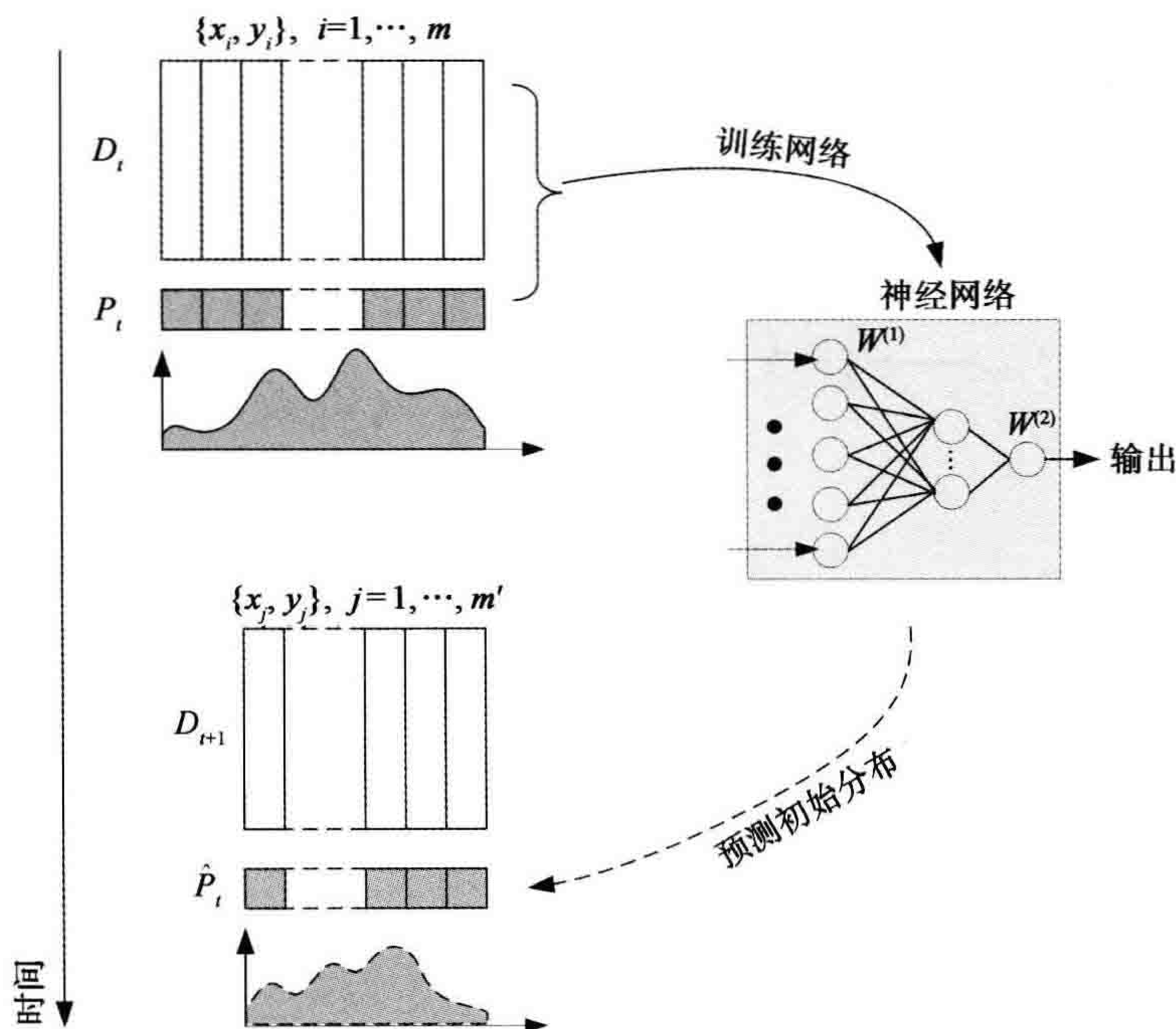


图 2-4 基于神经网络模型的映射函数

一般来说，这里使用具有多层感知(MLP)结构和反向传播算法的神经网络模型。基于数据信息、D_t 及其分布函数 P_t，我们可以设计一个神经网络模型以学习特征空间及其相应的数值加权函数 P_t 之间的关系。然后，对接收到的新数据集 D_{t+1}用经过训练的 MLP 来预测其分布函数的初始估计。反向传播(Werbos，1988，1990)是调整 $W^{(1)}$ 和 $W^{(2)}$ 的参数的关键(这里我们使用 $W^{(1)}$ 和 $W^{(2)}$ 分别代表 MLP 结构中“输入到隐层”和“隐层到输出”的权重)。因此，误差函数可定义为

$$e(k)=p_{(t)}(k)-o_{(t)}(k);\ E(k)=\frac{1}{2}e^2(k) \tag{2-10}$$

其中，k 代表反向传播训练时间点，$p_t(k)$、$o_t(k)$分别代表数据集 D_t 的分布函数的目标值、估计值。为了描述清楚，下文在 $W^{(1)}$ 和 $W^{(2)}$ 的更新规则推导中省略了下标(t)。

假设隐层和输出层均使用了 Sigmoid 函数，可定义 MLP 网络(见图 2-5)的相关公式如下。

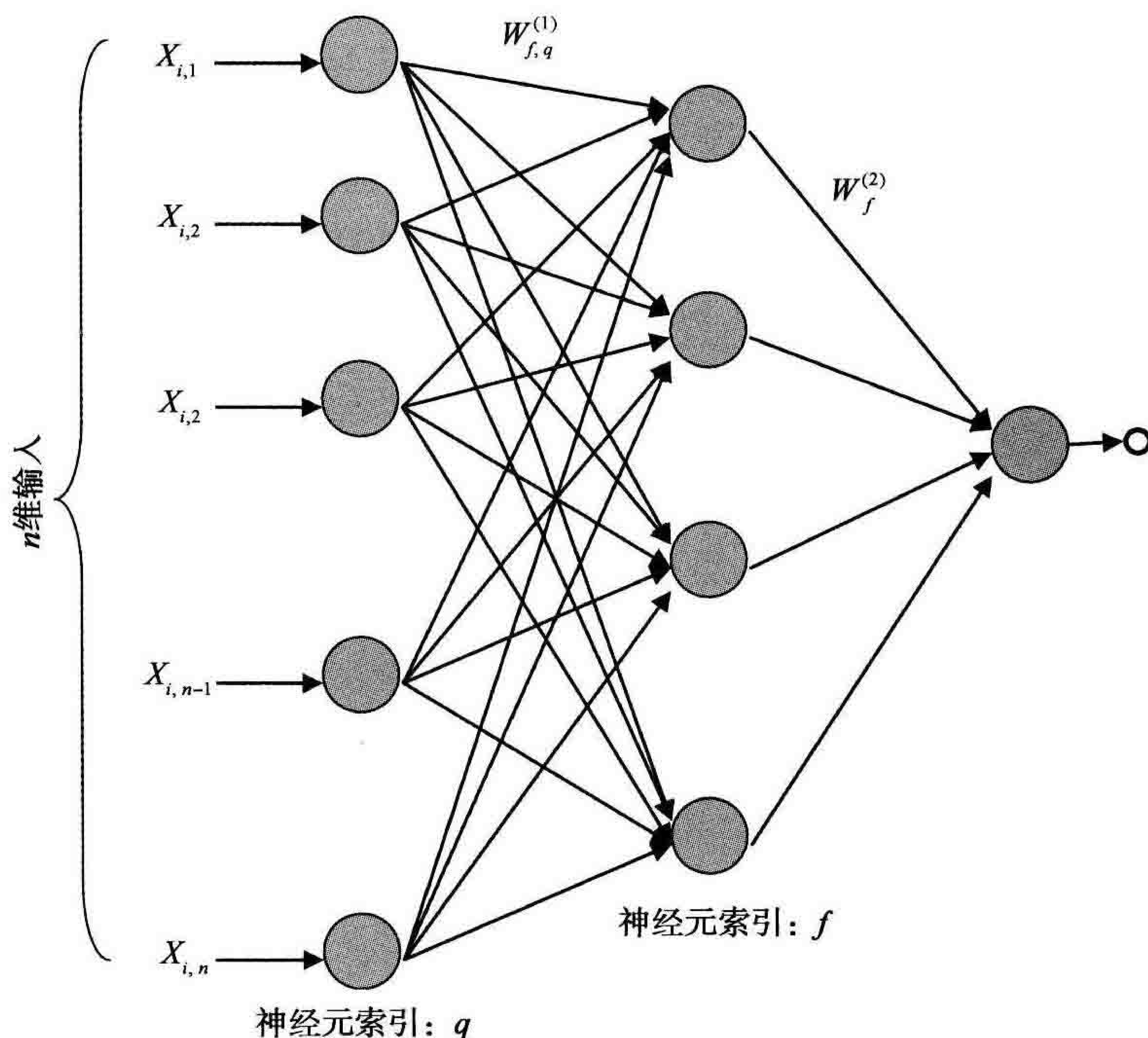


图 2-5 具有一个隐层的 MLP 非线性神经网络

$$o(k)=\frac{1-\exp^{-v(k)}}{1+\exp^{-v(k)}} \tag{2-11}$$

$$v(k)=\sum_{f=1}^{N_h} w_f^{(2)}(k)g_f(k) \tag{2-12}$$

$$g_f(k)=\frac{1-\exp^{-h_f(k)}}{1+\exp^{-h_f(k)}},\quad f=1,\cdots,N_h \tag{2-13}$$

$$h_f(k)=\sum_{q=1}^{n}w_{f,q}^{(1)}(k)x_{i,q}(k),\quad i=1,\cdots,N_h \tag{2-14}$$

其中，$h_f(k)$是第 f 个隐层的输入节点，$g_f(k)$是相应的输出，$v(k)$是 Sigmoid 函数前输出节点的输入，N_h 是网络中隐层神经元的数目，n 是总输入数。这里使用下标 f 和 q 分别表示隐层和输入层的神经元索引。

因此，可以用反向传播算法来更新神经网络的权值，从而学习特征空间与相应的分布函数之间的关系。这个过程可描述如下。

隐层到输出层的权重调整 $\Delta w_f^{(2)}$：

$$\Delta w_f^{(2)}=\alpha(k)\left[-\frac{\partial E(k)}{\partial w_f^{(2)}(k)}\right] \tag{2-15}$$

$$\frac{\partial E(k)}{\partial w_f^{(2)}(k)}=\frac{\partial E(k)}{\partial o(k)}\frac{\partial o(k)}{\partial v(k)}\frac{\partial v(k)}{\partial w_f^{(2)}(k)}=e(k)\cdot\frac{1}{2}(1-(o(k))^2)\cdot g_f(k) \tag{2-16}$$

输入层到隐层的权重调整 $\Delta w_{f,q}^{(1)}$：

$$\Delta w_{f,q}^{(1)}=\alpha(k)\left[-\frac{\partial E(k)}{\partial w_{f,q}^{(1)}(k)}\right] \tag{2-17}$$

$$\begin{aligned}\frac{\partial E(k)}{\partial w_{f,q}^{(1)}(k)}&=\frac{\partial E(k)}{\partial o(k)}\frac{\partial o(k)}{\partial v(k)}\frac{\partial v(k)}{\partial g_f(k)}\frac{\partial g_f(k)}{\partial h_f(k)}\frac{\partial h_f(k)}{\partial w_{f,q}^{(1)}(k)}\\&=e(k)\cdot\frac{1}{2}(1-(o(k))^2)\cdot w_f^{(2)}(k)\cdot\frac{1}{2}(1-g_f^2(k))\cdot x_{i,q}(k)\end{aligned} \tag{2-18}$$

其中，$\alpha(k)$是学习率，一旦 $W^{(1)}$ 和 $W^{(2)}$ 被调整，它就可以用来预测数据块 D_{t+1} 的初始分布函数 $\hat{P}_{t+1}$(对应于算法 2.1 中的式(2-1)和式(2-2))。这只须在 MLP 中基于新数据集 D_{t+1} 的特征空间做前馈传播。

2.4.3 基于在线评估系统的映射函数

本节用动态在线评估系统设计增量学习框架中的映射函数 ψ。引入三曲线拟合技术的新概念来鲁棒地预测式(2-1)和式(2-2)中的 $\hat{P}_t$ 值。

1. 三曲线拟合技术(TCF)

拟合函数的动态调整可以描述为一组基函数 $\varphi_i, i=1,2,\cdots,q$ 的线性组合，这里，q 是基函数的数量。其目标是从接收到的数据样本中动态地匹配值，以最小化所有数据 x 和 y 的最小均方误差(LSE)，如下式所示：

$$F = a_1 \times \varphi_1 + a_2 \times \varphi_2 + \cdots + a_q \times \varphi_q \tag{2-19}$$

这里 F 是一般目标值(特别情况下，初始分布函数值为 $\hat{P}_t$)。基函数的数量可根据精度要求和数据的噪声水平进行调节。

传统的单曲线拟合方法如图 2-6a 所示。在这种情况下，拟合的曲线并不能反映区域 B_1 和 B_2 中输入数据值的统计分布，这将导致这些区域的估计质量较差。可以从拟合曲线中计算近似数据的标准偏差(类似于误差线的概念)。但是这只给出了统计误差的均匀度量，并不能反映输入空间中不同区域近似值的不同质量(置信水平)。增大基函数的复杂度可能会“完美”拟合这些数据，但是会出现如图 2-6a 中虚线所示的过拟合问题。

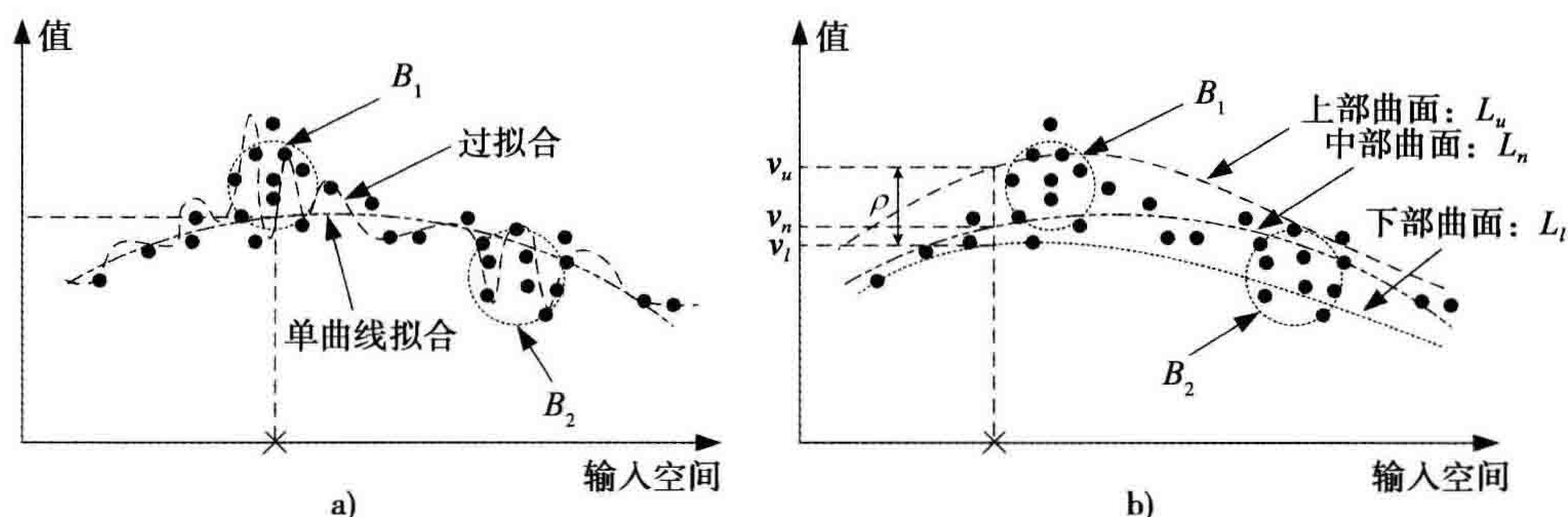


图 2-6　单曲线拟合与三曲线拟合

为了克服这一限制，He & Starzyk(2007) 提出了三曲线拟合(TCF)技术，如图 2-6b 所示。这种方法引入三条曲线，分别为中部曲线(L_n)、上部曲线(L_u)和下部曲线(L_l)，用来拟合不同空间的数据样本：

- 中部曲线：拟合输入空间的所有数据样本，与图 2-6a 所示的曲线一样。
- 上部曲线：仅拟合中部曲线之上的所有数据样本。
- 下部曲线：仅拟合中部曲线之下的所有数据样本。

从图 2-6b 可以看出，中部曲线 L_n 给出了拟合值的粗略估计，而上部曲线 L_u 和下部曲线 L_l 给出了局部统计分布信息或不同输入空间的置信估计，因此，可以局部描述中部曲线估计值的统计偏差，如图 2-6b 中的 ρ 值所示。ρ 值是由 v_u 和 v_l 确定的($\rho = |v_u - v_l|$)，而 v_u 和 v_l 的估计值分别是由 L_u 和 L_l 曲线确定的。这样，通过中部曲线和它的真值相比较，ρ 值反映了估计值 v_n 的置信水平：小的 ρ 值意味着估计值 v_n 有较好的置信度。因此，对于有 k 个处理单元的评估系统，在每一个系统的输入空间执行 TCF 方法(详细内容请参见本节后面的“在线评估估计的系统级架构”小节)，投票机制由以下公式实现：

$$v_{vote} = \frac{\sum_{i=1}^{k}(v_{ni}w_i)}{\sum_{i=1}^{k}w_i} \tag{2-20}$$

其中，每个处理单元的投票权重定义为 $w_i = \frac{1}{\rho_i}$，$i=1,\cdots,k$。

为了利于增量学习，必须对这些三曲线进行动态调整以适应新数据。通过这种方式，随着时间收到的任何新数据样本仅在必要时修改其相应的曲线，因此以前的知识将被保留。图 2-7 和对应的算法 2.2 说明了这一方法。当出现一个新数据样本时，首先修改 L_n，然后计算拟合值 v_n 和真值 v 之间的差值 d。如果 d 小于 0，修改 L_u，L_l 保持不变；否则，修改 L_l，L_u 保持不变。

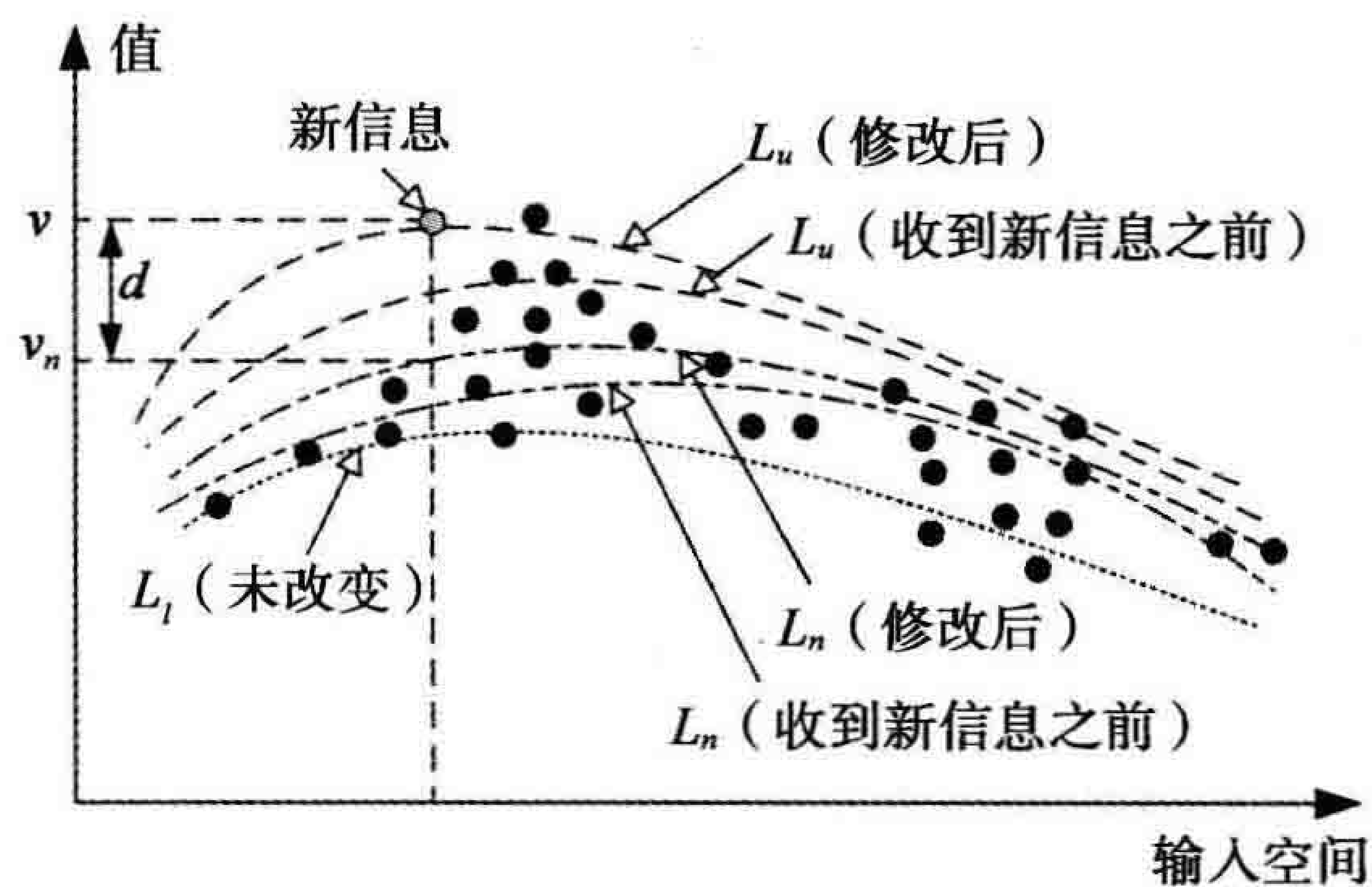


图 2-7　实现 TCF 的动态学习

[算法 2.2]　TCF 学习

1）出现新数据。

2）动态地修改 L_n。

3）计算 $d=v_n-v$。

4）如果 $d<0$，则

　　修改上部曲线 L_u；

　　保持下部曲线 L_l 不变。

否则，

　　修改下部曲线 L_l；

　　保持上部曲线 L_u 不变。

结束。

这种动态学习方法需要在线估计式(2-19)中的系数 $a_1,\cdots,a_q$，因此，有必要累积基函数值及其对于不同输入数据的组合。为此，式(2-19)可表示如下：

$$F=[\varphi_1\varphi_2\cdots\varphi_q]\begin{bmatrix}a_1\\a_2\\\cdots\\a_q\end{bmatrix}=\Phi\cdot A \tag{2-21}$$

逼近函数的系数可从下式获得：

$$\begin{bmatrix}a_1\\a_2\\\cdots\\a_q\end{bmatrix}=(\Phi^T\Phi)^{-1}\Phi^T F$$

$$=\begin{bmatrix}\sum_{i=1}^{n}\Phi_{1i}\Phi_{1i} & \sum_{i=1}^{n}\Phi_{1i}\Phi_{2i} & \cdots & \sum_{i=1}^{n}\Phi_{1i}\Phi_{qi}\\ \sum_{i=1}^{n}\Phi_{2i}\Phi_{1i} & \sum_{i=1}^{n}\Phi_{2i}\Phi_{2i} & \cdots & \sum_{i=1}^{n}\Phi_{2i}\Phi_{qi}\\ \vdots & \vdots & \cdots & \vdots\\ \sum_{i=1}^{n}\Phi_{qi}\Phi_{1i} & \sum_{i=1}^{n}\Phi_{qi}\Phi_{2i} & \cdots & \sum_{i=1}^{n}\Phi_{qi}\Phi_{qi}\end{bmatrix}\cdot\begin{bmatrix}\sum_{i=1}^{n}\Phi_{1i}F_i\\ \sum_{i=1}^{n}\Phi_{2i}F_i\\ \vdots\\ \sum_{i=1}^{n}\Phi_{qi}F_i\end{bmatrix} \tag{2-22}$$

这里 n 是数据点的个数。在线实现要求存储式(2-22)中 $s=\frac{q(q+1)}{2}+q$ 个不同的组合值。

$$\begin{cases}\sum_{i=1}^{n}\Phi_{ki}\Phi_{mi}\\ \sum_{i=1}^{n}\Phi_{ki}F_i\end{cases}\quad 其中\ k,m=1,\cdots,q \tag{2-23}$$

在接收到新数据时，更新 s 的值，并用新系数 a_1，…，a_q 计算式(2-22)。通常，对于 q 个基函数，需要存储 s 个组合和 $q\times q$ 的逆矩阵($\Phi^T\Phi$)以更新逼近方程的系数。

2. 在线评估估计的系统级架构

基于 TCF 方法，图 2-8 给出了所提出的评估系统中估计式(2-1)和式(2-2)中 $\hat{P}_t$ 的系统级架构。该系统有两个网络结构：数据处理网络(DPN)和信息处理网络

(IPN)。DPN 负责输入数据空间变换和在线动态数据拟合；IPN 负责 DPN 所提供的结果的最终投票。每一个数据处理单元(DPE)都将执行本节前面的“三曲线拟合技术(TCF)”小节所讨论的三曲线拟合，并为信息处理单元(IPE)输出 v_{ni}、v_{ui} 和 v_{li} 的拟合值。这些值提供了真值的粗略估计以及统计分布信息。基于该信息，每个 IPE 将根据式(2-20)对最终值进行投票。

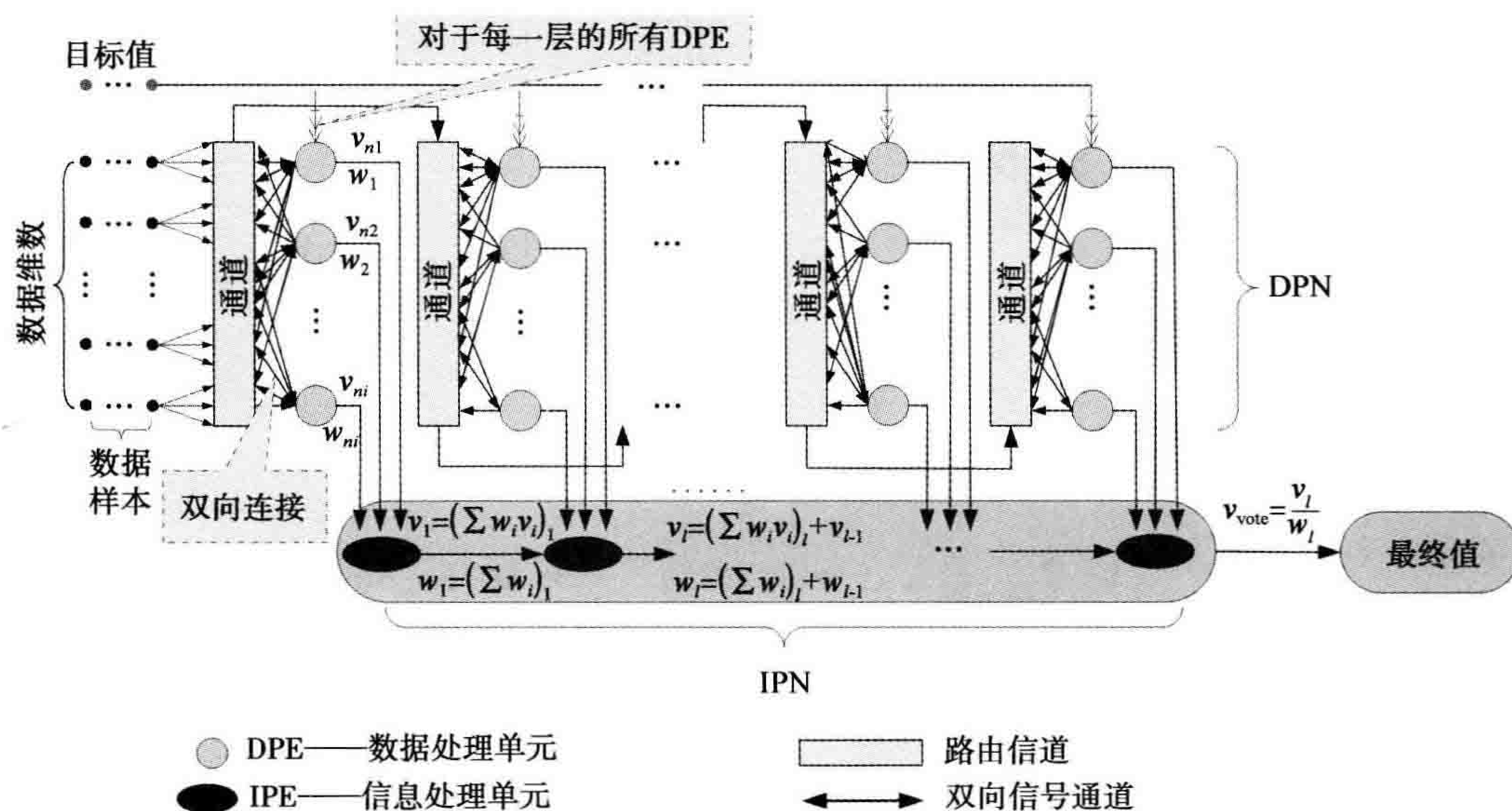


图 2-8 增量学习在线动态评估系统

这种架构对信息的处理方式类似于管线式移位寄存器结构。每个 DPE 都有一组输入，并伪随机地连接到本地路由信道。在第一个时钟周期中，数据在第一层信道中有效，第一层 DPE 将读取这些数据作为它们的输入。处理之后，这些单元将把变换得到的数据输出到输入信道的相同位置。同时，在 IPN 网络中，也将输出它们的估计值 v_{ni} 及其相应的权重 w_i 到 IPE。在下一个时钟周期，IPE 将根据下面的公式将这些局部值及其权重相结合并传递给 IPE 的下一层。

$$v_l = (\sum w_i v_i)_l + v_{l-1} \tag{2-24}$$

$$w_l = (\sum w_i) + w_{l-1} \tag{2-25}$$

这里，下标“l”表示信息来自信道层 l。因此，v_l 和 w_l 是 l 层的组合信息和权值。

当到达下一时钟周期时，转换的数据(DPF 在第一层信道的输出数据)被转移到下一层信道中作为 DPE 在第二层信道的输入数据，而另一组输入样本可以被发送到第一层信道。与此同时，第二层的 IPE 是该层信息与先前层转换后信息的组合。

在所有时钟周期中，系统中所有的处理单元是激活的，从而使该架构适用于动

态在线学习。最后，当数据到达最后一层时，计算最终投票值。在式(2-1)和式(2-2)中，这些值将用于 $\hat{P}_t$ 值的初始估计。

为了详细分析各个处理单元的组织和操作，图 2-9 显示了 DPE 的局部组织和连接构造。

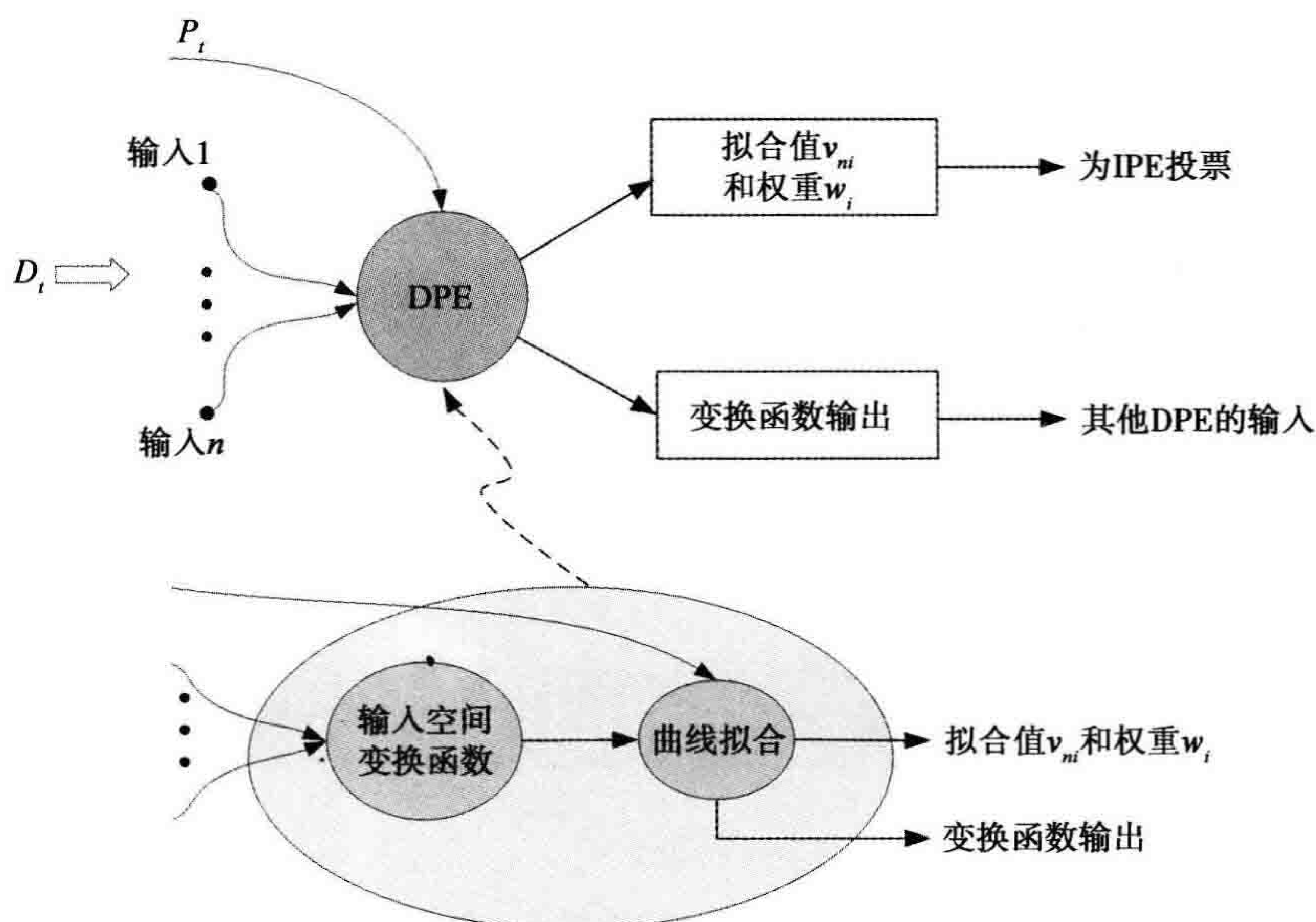


图 2-9 DPE 的详细结构

在训练阶段，对于时间 t，输入数据 D_t 和相应的目标值 P_t 作为本地信道的输入。每一个 DPE 利用一组输入空间变换函数来组合源自不同输入空间的信息。根据 TCF 技术，每个 DPE 将动态地修改其拟合曲线的系数。每个 DPE 还将通过输入空间变换函数输出数字值，该输出将被作为下一层 DPE 的输入。训练之后，每个 DPE 拥有由 D_t 决定的 TCF 系数。

每当下一数据块 D_{t+1} 有效时，每个 DPE 将根据训练获得的拟合曲线输出预测值 v_{ni} 和相应的权重 w_i(从 v_{ui} 和 v_{li} 计算所得)。所有的这些信息将作为对 D_{t+1} 的分布函数的初始估计值(对应于式(2-1)和式(2-2))，在 IPN 中参与对 $\hat{P}_t$ 值的投票(见式(2.20))。

通过比较 2.4.1 节、2.4.2 节和 2.4.3 节所描述的三种映射函数，可以预见，与简单的欧氏距离法相比，基于映射函数的回归模型(如非线性神经网络模型)和评估系统可以更好地估计分布函数 $\hat{P}_t$。另外，回归模型和评估系统所需要的计算开销高于欧氏距离映射。对于不同的应用场景，除了具体需求之外，映射函数的选择还需要权衡性能与计算开销。此外，需要指出的是，其他方法也可以用来设计这类映射函

数。例如，也可用多维概率密度估计方法来估计分布函数(Silverman，1996；Scott，1992)。一般来说，P 的有效估计(图 2-2 中的第二层)对增量学习框架至关重要，这也是一个活跃的研究课题。希望上述三种方法能够成为不同应用问题的有用选择。

2.5 实例研究

本节用两个实例研究说明自适应增量学习框架的应用。第一个例子是视频流数据的学习，第二个例子是互联网垃圾邮件预测。这两个应用实例表明增量学习框架能随着时间从连续数据中适应性地学习和累积经验，并使这些知识有利于后续的预测和决策过程。

2.5.1 视频流的增量学习

首先介绍增量学习框架在针对多目标学习和场景分析的视频流数据分析中的应用(He & Chen，2008；He，Chen，Cao，Desai & Hohil，2008；He，Chen，Cao & Starzyk，2008)。多目标学习和场景分析在感知、推理、行动和任务导向的行为分析方面具有重要的地位，因此认为它是机器智能研究中一个重要问题(He 等，2008；Can & Grossberg，2005；Grossberg，1999；Grossberg & Howe，2003；Wang，2005)。

1. 特征表示

在应用增量学习框架之前，为了学习和预测，需先表示视频流中不同物体(类别)。目前已有很多表示方法，例如，由图像中的局部关键点所表示的尺度不变特征变换(SIFT)(Lowe，1999，2003；Ke & Sukthankar，2004；Zickler & Veloso，2006)。包括阈值法、基于边缘的方法、基于区域的方法及连通不变性松弛法等的各种图像分割方法也可用于提供此类表示方式。这里，先将原始图像转换为灰度图像，然后基于边缘进行分割，随之用膨胀、形态学填充、纹理消除来识别每个潜在物体的质心和边界框。图像分割可详见文献 Gonzalez & Woods(2002)。

由于不同的物体在经过特征表示步骤后大小不同，为了便于训练和测试，所有的物体(边界框定义的区域)被缩放到相同的大小，如图 2-10 所示。在此过程中，首先寻找训练数据中所有潜在物体的最小高度 $H_{\min}$ 和最小宽度 $W_{\min}$($H_{\min}$ 和 $W_{\min}$ 可以是也可以不是由同一个对象决定的)，然后将所有边界框缩放到相同的尺寸(He & Chen，2008)。

$$\frac{H}{H_{\min}} = A_H + \mathrm{residual}_H \tag{2-26}$$

$$\frac{W}{W_{\min}} = A_W + \text{residual}_W \tag{2-27}$$

其中，residual_H 和 residual_W 是除法的余数。为了保留缩放信息，残差可以是 A_H 和 A_W 之间的随机分布：

$$A_h = A_H + \text{rand}[0 \quad 1] \tag{2-28}$$

$$A_w = A_W + \text{rand}[0 \quad 1] \tag{2-29}$$

这样，在所有矩形区域中所包含的像素的数目不会大于 1。

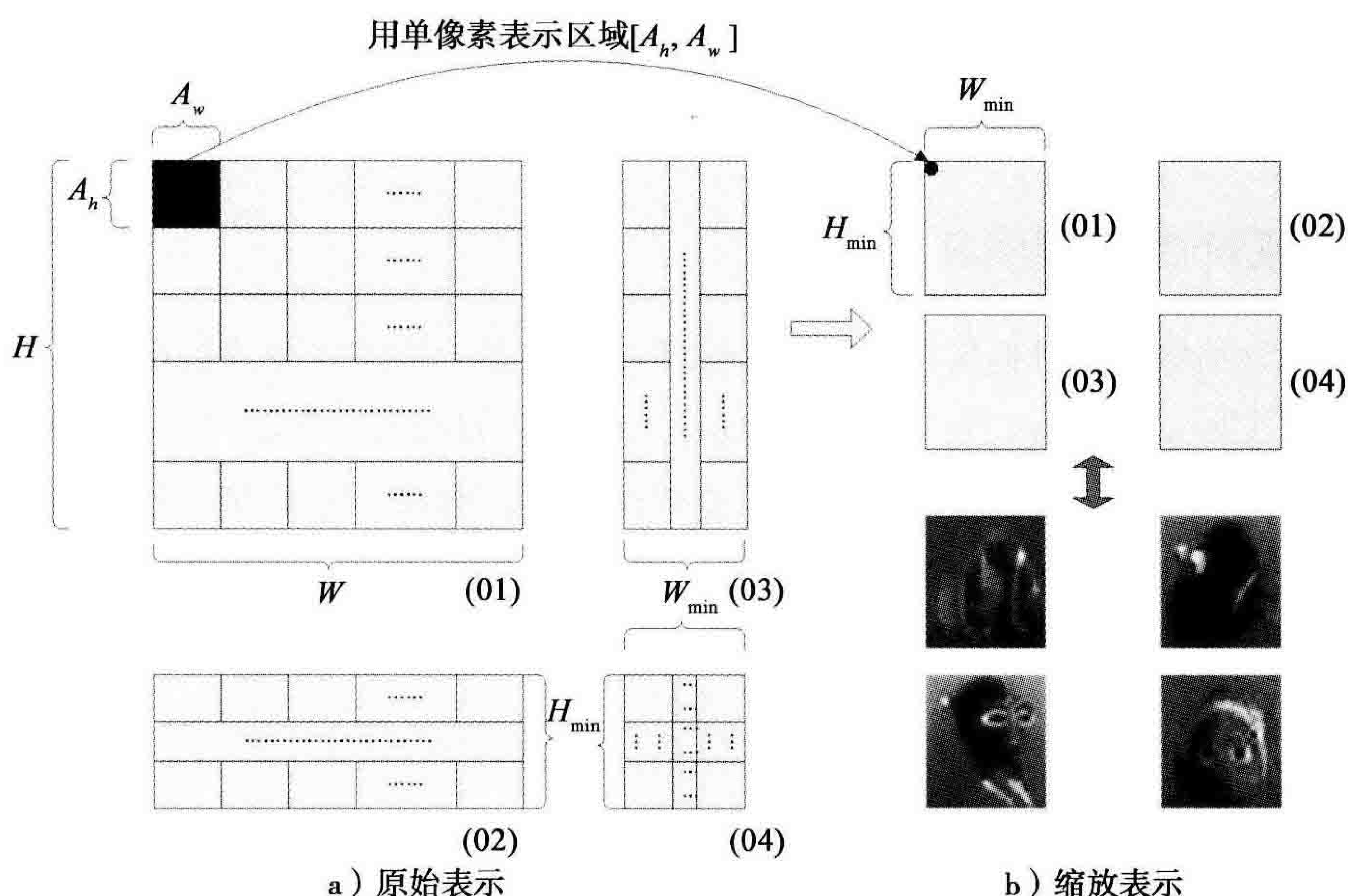


图 2-10　特征表示

2. 实验结果

用从 YouTube(*www. youtube. com*，2009)抓取的不同视频数据来测试增量学习框架。第一个视频片段“Finding Nemo”含有两类物体：Dory 和 Marlin，分别表示为“D”和“M”。根据“特征表示”小节中介绍的数据处理方法，总共提取了 4000 个图像数据样本。每个数据样本由 600 维特征向量表示。随机选择 2000 个样本以用于训练，剩余的 2000 个样本用于测试性能。同时，假设训练数据以 100 个块递增，每块包含 20 个样本。图 2-11 显示了该应用中增量学习架构的数据流(He & Chen，2008)。

在这个实例中，用含有一个隐层的 MLP 结构神经网络作为基本学习算法。隐层神经元的数量设定为 10，输入神经元、输出神经元分别等于特征数量和类别数量，用 Sigmoid 函数作为激活函数，用迭代 500 次的反向传播来训练神经网络模型。在当前仿真中，使用 2.4.1 节讨论的欧氏距离映射函数。

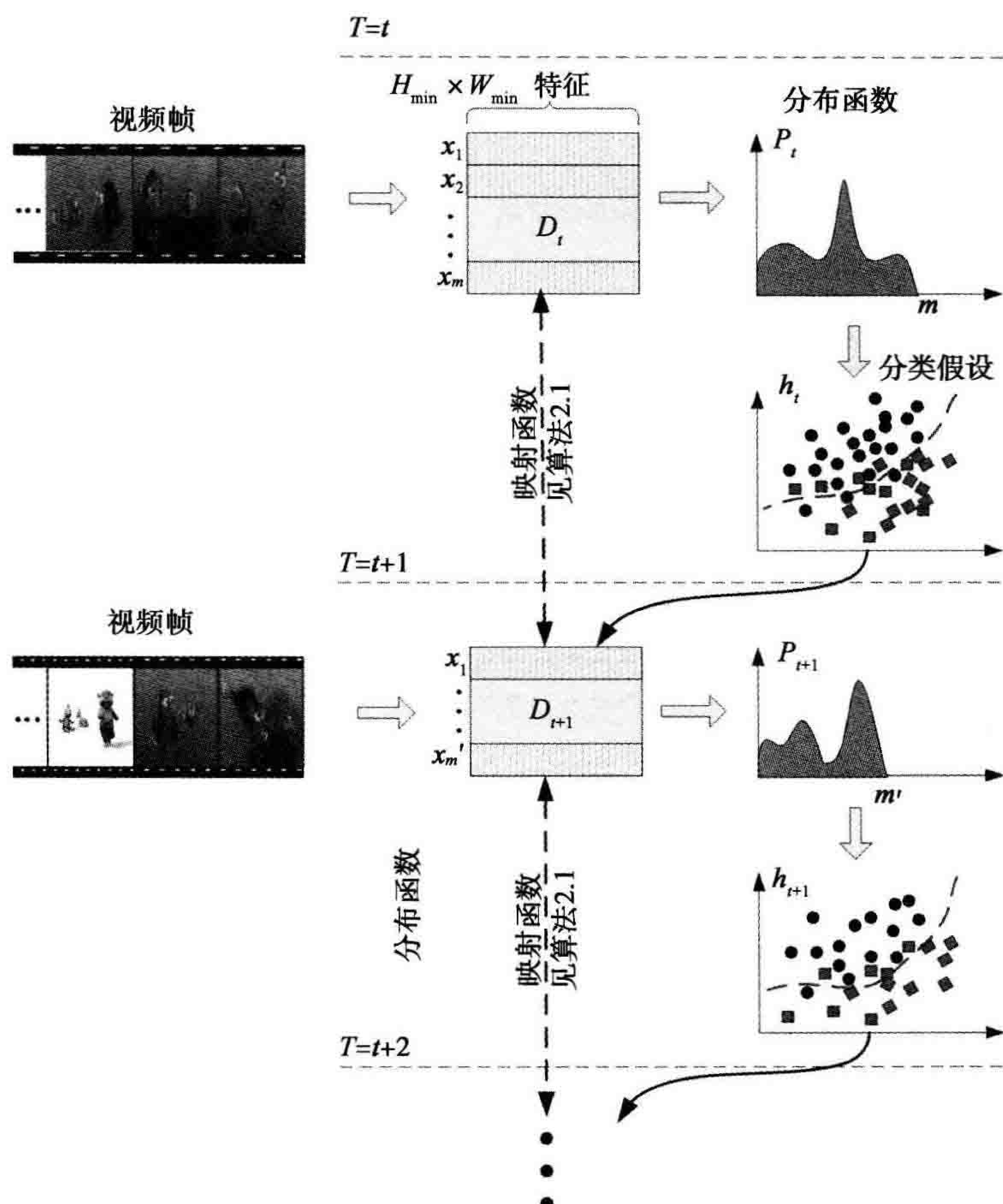


图 2-11　视频数据分析的增量学习

图 2-12 显示了学习过程的识别误差性能(基于随机运行 20 次的平均值)(He & Chen，2008)。这里，用对所有测试样本的正确识别率来度量不同学习阶段的误差率。图 2-12 也显示了在不同学习阶段运行了 20 次的误差条形图信息，误差条形图的下降趋势表明随着知识累积，识别结果更加稳定和鲁棒。

3. 增量学习中的概念迁移问题

概念迁移是增量学习中的一个重要问题，例如，在多对象学习时，在学习过程中时常引进新的感兴趣对象。考虑 2.3 节中的增量学习框架，假设在 $t+1$ 时刻，接收到的训练数据可以表示为 $D_t=(D_s, D_n)$，其中 D_s 表示先前观测到类别的样本，D_n 表示学习系统到目前为止还没有学习的样本(新对象)。这种情况下，D_n 中的样本将被分类假设 h_t 错误分类。因此，根据式(2-3)和式(2-4)，对于新类别(概念)的权重将会增加。以这种方式，提出的增量学习框架可以为新引进的类别样本自动分配更高的权重，以便从新对象的信息中积极地学习。

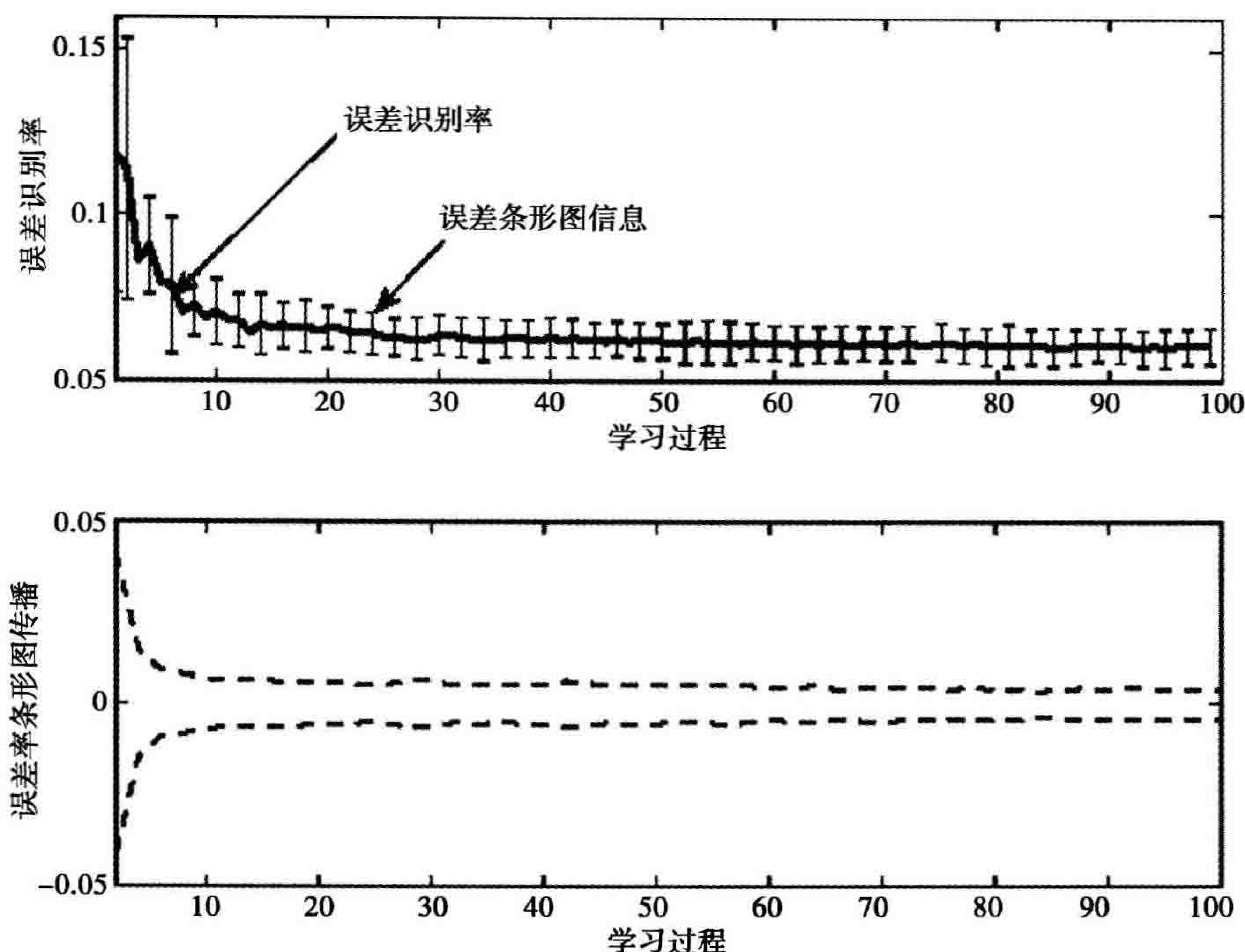


图 2-12 识别误差的降低(测试阶段)

为了测试概念迁移问题，我们组合视频数据“Finding Nemo”和一个新的视频剪辑“Baby Shrek.”。在“Baby Shrek”中有 3 类物体：Shrek Jr. 1、Jr. 2 和 Jr. 3，分别表示为 J1、J2 和 J3。在整个学习过程中，用两种学习场景对系统进行训练。对于场景 1，在整个学习过程只使用“Finding Nemo”中的图像进行训练。对于场景 2，在第 30 块处引入新对象(“Baby Shrek”)，也就是说，$T1$ 期(第 1～29 块)只包含“Finding Nemo”中的图像，而 $T2$ 期(第 30～100 块)包含两个视频数据流中的图像。在这两种情况下，测试数据中的一半来自一个视频剪辑，另一半来自另一个视频剪辑。图 2-13 显示了测试数据对应于学习过程周期的识别误差率，其中实线表示场景 2，虚线表示场景 1。从理论上讲，场景 1 的最小误差识别率被限制在 50%。这是因为没有一个“Baby Shrek”对象(一半的测试数据)可以被学习系统正确识别，因为在整个学习过程中从来没有学习过这些信息。另一方面，对于场景 2，在 30 块处引入新概念时，误差识别率开始逐渐降低。这种改善是因为增量学习方法可以自动地学习新对象的信息，并且能够使用获得的知识提高对测试数据的识别性能。

为了观测所提出的增量学习框架如何针对概念迁移问题自适应地调整分布函数，图 2-14 显示了从时间 t_{29} 到 t_{30} 权重改变的详细视图，这意味着仅在时间 t_{30} 处引入新概念前后，所有的这些新目标(概念)趋向于获得比旧目标更高的权重。这促使系统积极地从新概念中学习知识，所以对于新知识是自适应的(He & Chen，2008)。

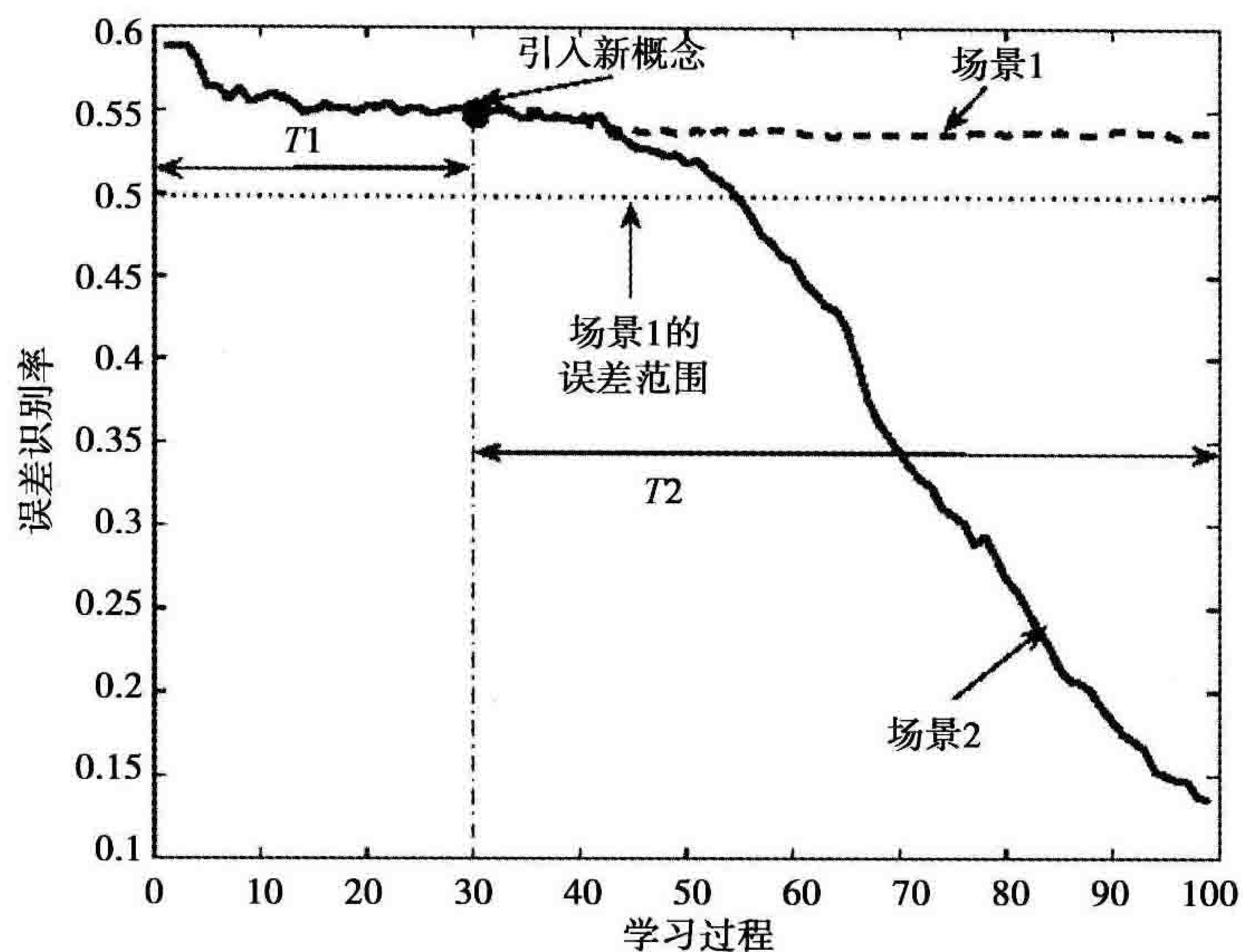


图 2-13 概念迁移：在第 30 块处引进新概念

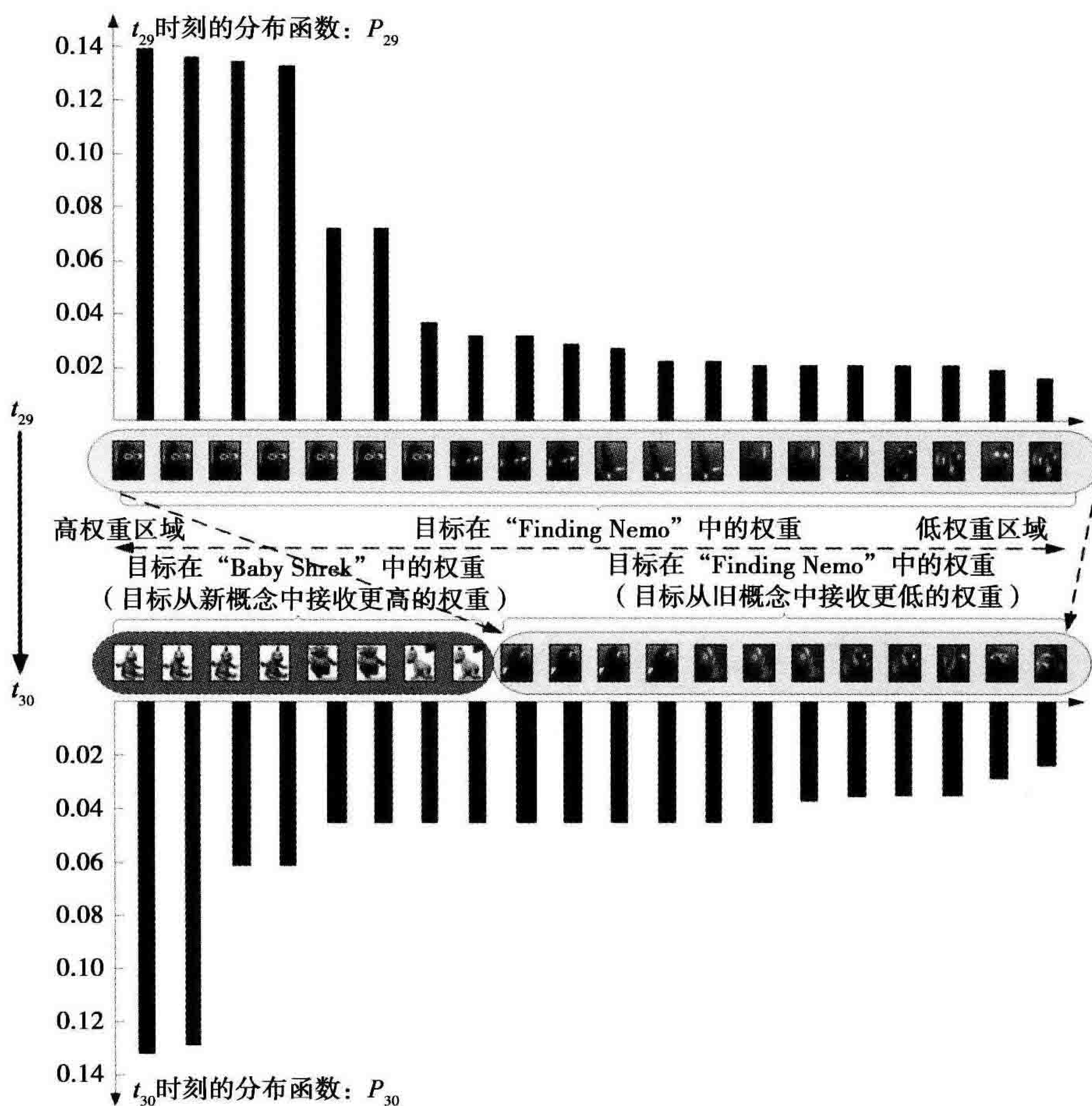


图 2-14 对于概念迁移分布函数的自适应调节

在增量学习过程中关于概念迁移的另一个有趣问题是系统对于新概念信息的自我调节速度。从图 2-13 可以看到，虽然在时间 t_{30} 处引入了新概念，但在这个时间点，系统识别的误差仍不能显著减少。这是因为此时系统会基于 $T1$ 期内前 29 块训练数据的积累知识加上第 30 块新信息的知识做出决策。随着 $T2$ 期内对新概念的不断学习，系统基于新知识逐渐提高了识别性能。这反映出误差率随着学习时间逐渐减少的特点(学习场景 2)。就如同人类大脑的高级智能：一个在某个领域有广泛知识的人，在最初可能会抵触新概念信息，他或她可能更加喜欢根据以前已经建立的经验做出决策。但随着不断探索新概念和知识，他或她会逐步学习，并使用所获得的新信息帮助决策。

为了测试这个假设，图 2-15 给出了另外一种情况，在第 50 块处引入新概念。在这种情况下，因为新概念($T2$ 期)的学习过程比图 2-13 中的短，所以最终识别的误差比图 2-13 中的高。这就提出了类脑智能研究中一个有趣的基本问题：何时是增量学习过程引入和学习新概念的最佳时间(He & Chen，2008)。

通过比较图 2-13 和图 2-15，可以看出，系统在第 30 块处引入新概念比在第 50 块处引入新概念可以获得更高的识别率(更低的识别误差)。这意味着，引入新概念的阶段越早，系统学习和使用这些知识实现目标的机会越大。然而，在真实学习环境中，这可能是困难的，甚至是不可能的(He & Chen，2008；He 等，2008)。原因如下：第一，某种类型的知识可能仅在人类生命的中间阶段出现。第二，在发育的早期阶段，大脑资源(如记忆)和功能非常有限，例如，不能期望一个 1 岁的婴儿

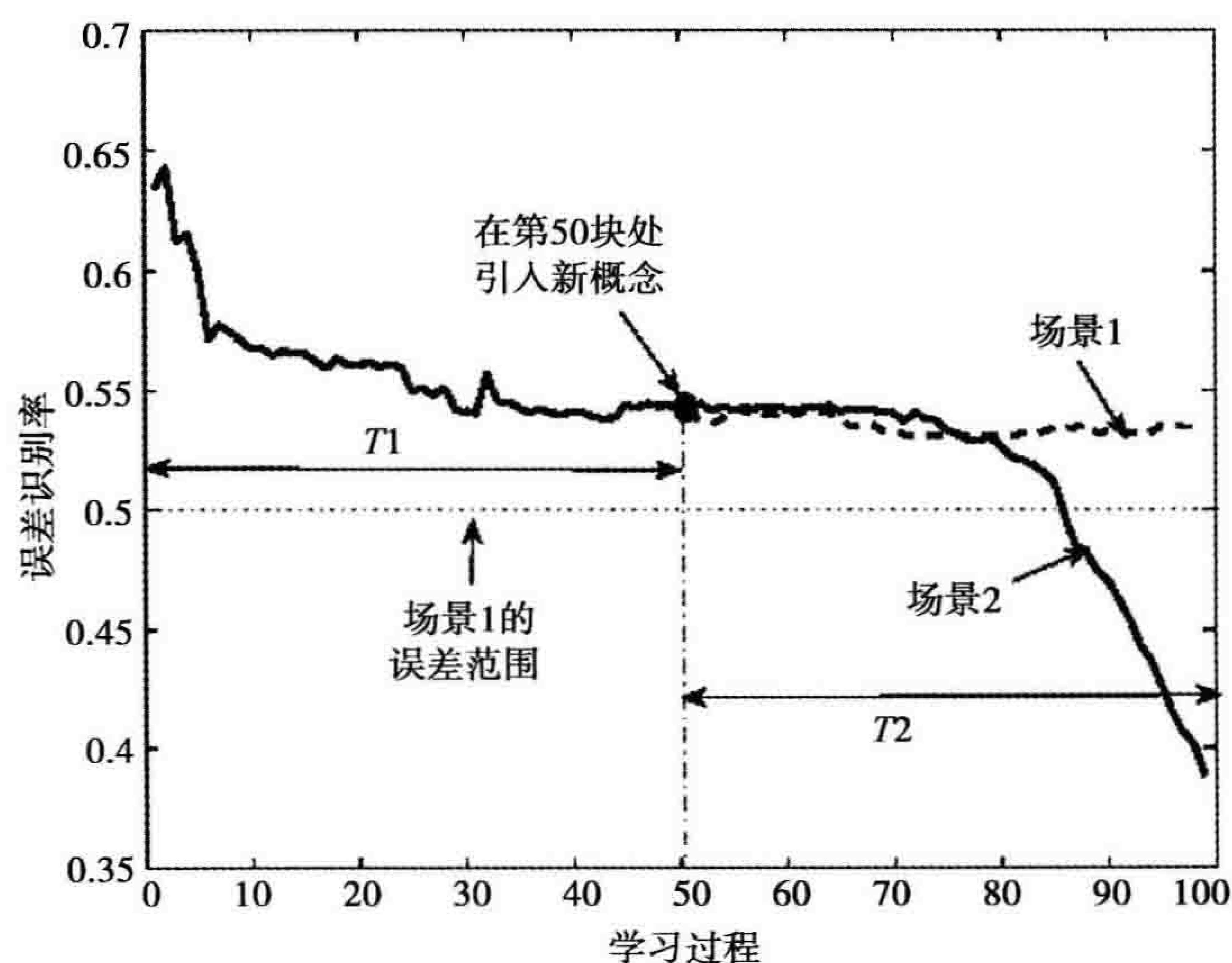


图 2-15　概念迁移：在第 50 块处引进新概念

记住所有的中文或英文词汇。因此，资源限制了学习能力。第三，现实的学习场景是与外部环境主动交互的，在学习过程中不可避免地要面对新概念。另一方面，在学习过程中，如果新概念引入太晚，智能系统可能没有足够的时间(对应于一个短暂的 $T2$ 期)去探索和完全掌握这些知识。

2.5.2　垃圾邮件分类的增量学习

该实例说明了所提出的框架在预测垃圾电子邮件方面的增量学习能力。最近，涉及电子邮件通信和垃圾邮件的安全问题越来越多，这引起了学术界和工业研究机构的极大关注。例如，根据 400 个网络安全专业人员的调查，垃圾邮件已经超过病毒成为主要的有害网络入侵(Whitworth & Whitworth，2004)。此外，根据 Ferris Research 的"垃圾邮件的全球经济影响"(2005)报告，据估计，垃圾电子邮件导致的生产力下降和其他损耗使美国企业的损失从 2003 的 100 亿美元增加到 2005 年的 170 亿美元，全球总损失达 500 亿美元(Ferris Research，2005)。一些现有的区分垃圾电子邮件的技术包括基于支持向量机(SVM)的方法(Drucker & Vapnik，1999)、神经网络方法(Yang & Elfayoumy，2007)、粒子群优化算法(PSO)(Lai & Wu，2007)和贝叶斯信度网络(Zhang & Li，2007)等。这个实例的目的是说明增量学习框架可以适应性地学习连续收到的电子邮件数据，并随着时间积累知识，提高对垃圾邮件的检测性能。

1. 数据集特性和系统配置

这个实例使用了 UCI 机器学习知识库的垃圾邮件数据库(Asuncion & Newman，2009)。这个数据库包含 4601 封电子邮件，其中 2788 封合法邮件，1813 封垃圾邮件。每封电子邮件用 57 个属性表示，其中 48 个属性编码特定词的频率(FW)，6 个属性编码特定字符的频率(FC)，3 个连续属性反映了电子邮件中大写字母的统计(SCL)信息，即连续大写字母序列的平均长度、连续大写字母序列的最大长度和电子邮件中大写字母的总数量。表 2-1 列出了测试电子邮件数据的属性特征。

如 2.4.1 节和 2.4.2 节所讨论的那样，分别使用欧氏距离模型和神经网络模型来设计映射函数。对于神经网络映射函数，隐层神经元的数量设定为 10，输入神经元和输出神经元的数量分别设为 57 (属性的数量)和 1。使用 Sigmoid 函数作为激活函数，并用反向传播训练神经网络。将学习率设为 0.1，进行 1000 次训练迭代，采用决策树方法作为基本分类算法。

为了显示统计信息，所有展示的结果都是 100 次随机运行的平均值。在每次运行中，随机选取一半的电子邮件作为训练数据，剩下的一半作为测试数据。此外，均分训练数据为 20 块，并且假定它们随时间有效递增。

表 2-1 电子邮件数据属性特征

<table>
<tr><th>词频(FW)：特定的词在电子邮件中所占的百分比</th><th>字符频率(FC)：特定的字符在电子邮件中所占的百分比</th><th>大写字母统计(SCL)</th></tr>
<tr><td>make remove people you hp telnet 1999 original</td><td>;</td><td rowspan="6">连续大写字母序列的平均长度；连续大写字母序列的最大长度；电子邮件中大写字母的总数量</td></tr>
<tr><td>address Internet report credit hpl 857 parts project</td><td>(</td></tr>
<tr><td>all order address your george data pm re</td><td>[</td></tr>
<tr><td>3d mail free font 650 415 directedu</td><td>!</td></tr>
<tr><td>our receive business 000 lab 85 cs table</td><td>$</td></tr>
<tr><td>over will email money labs technology meeting conference</td><td>#</td></tr>
</table>

2. 仿真结果

图 2-16 显示了增量学习框架在学习过程中的垃圾邮件分类误差率。可以看到，随着学习过程的推进，分类性能在逐步提高，这表明提出的方法能够随着时间推移有效地学习和积累经验，并使用这些知识促进未来从新数据中学习。

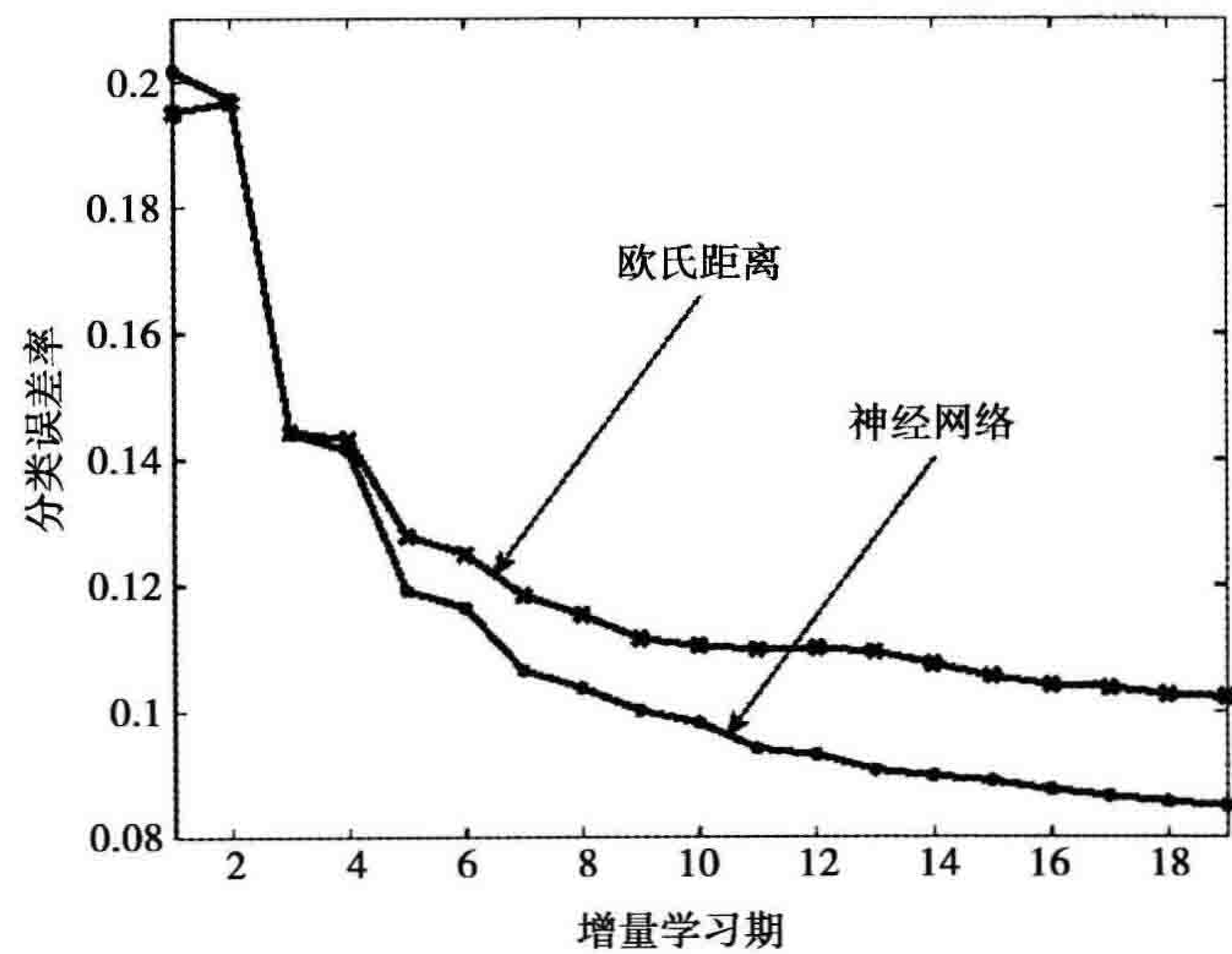


图 2-16 垃圾电子邮件分类误差率与增量学习(欧氏距离模型和神经网络模型)

可以看出，对于这些数据，神经网络映射函数比欧氏距离映射函数具有更好的检测性能。为了更清楚地解释这一事实，用 ROC 图进一步分析(Fawcett，2003，2006)。ROC 曲线是常用的基于分类混淆矩阵的评价方法，是以真阳性率(TP 率)为

纵坐标、假阳性率(FP率)为横坐标绘制的曲线。ROC空间中的任何一点对应于给定分布的单一分类器的性能。一般来说，对于硬分类器，只能输出预测类标签，每个标签对应ROC空间的一个点(FP率、TP率)。另一方面，软分类器，如神经网络，其输出样本属于每个类的似然，用ROC空间的曲线表示。ROC曲线通过调整决策阈值去生成一系列的点而绘制成的。软分类器的性能可通过计算其ROC曲线下的面积(AUC)来衡量。需要指出的是，基于对分类器的内在特征的观测，通常可以直接使硬分类器输出软分类结果输出(Freund & Schapire，1996)。

图2-17展示了用所提出的方法对两种类型的映射函数的ROC分析。仿真实验中执行了100次随机运行，并绘制出了平均ROC曲线。平均ROC曲线的获得方法如Fawcett(2003)所讨论的那样：固定FP率，取ROC曲线的垂直样本，计算对应的TP率的平均值。从图2-17可以看出，基于神经网络映射函数学习方法的ROC曲线比基于欧氏距离函数学习方法的ROC曲线高。这意味着，在这个实例中，非线性神经网络映射函数具有更好的分类性能。但也应该注意到，神经网络映射函数比欧氏距离函数需要更多的计算时间。根据当前的仿真实验环境(Intel Core 2 Duo CPU E4400 @ 2.00 GHz、2.0 GB RAM 和 Matlab 7.2.0.232 版本(R2006A))，神经网络映射函数需要运行68.7894s，而欧氏距离映射函数仅需要运行0.0177s。选择映射函数时，可参考这些结果，以在应用需求、精度和计算成本之间取得折中。

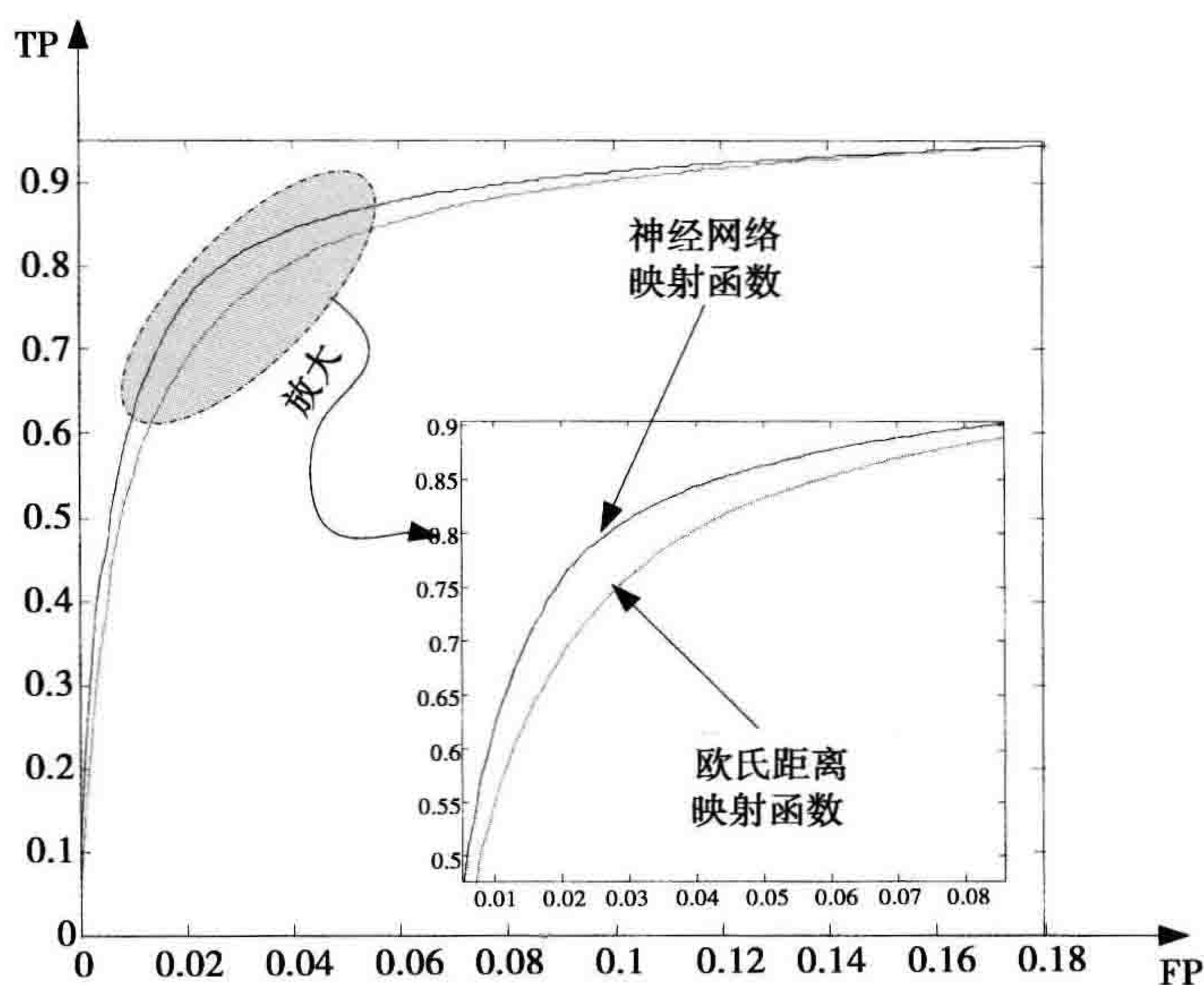


图2-17 不同映射函数的ROC分析

2.6 总结

- 无论从内因还是外因来看，增量学习对于理解类脑智能并使这种水平的智能更加接近现实都是至关重要的。就自身而言，智能系统应该能在其生命周期内不断增量学习，随着时间积累经验，并会使用这些知识实现后续任务。从外部看，智能系统与环境交互产生的原始数据在无限(或有限)长的学习周期内不断增量涌现，因此智能系统应该能够自适应地用这些连续数据实现增量学习。
- 基于自举和自适应学习原则，本章给出了一个通用的增量学习框架。该框架的目标是将以前学到的知识用于当前接收到的数据，使其有助于从新信息中学习知识，并随着时间不断累积经验，最终实现学习周期内的全局泛化。
- 映射函数的设计对于增量学习框架是至关重要的。本章讨论了 3 种设计方法：欧氏距离方法、回归学习方法、在线评估系统方法。
- 在增量学习期间，概念的漂移对于理解其鲁棒性和学习能力是至关重要的。例如，在场景分析的学习中会出现新对象。因此，一个智能系统应该有能力自动调整知识库，从而学习新知识。本章讨论了两个问题，第一，在增量学习框架中使用什么样的机制使得智能系统能够自适应地调整以适应新概念？第二，一个学习系统应如何快速转移它的决策边界和知识库以适应新概念？
- 增量学习在不同领域内具有广泛的应用。本章视频数据流和垃圾电子邮件数据的实验结果证明了该方法的有效性。这两个实例的研究为如何使用增量学习框架解决实际应用问题提供了有用的建议。

参考文献

Asuncion, A., & Newman, D. J. (2009). UCI machine learning repository *[Online]. Available: http://archive.ics.uci.edu/ml/datasets.html.*

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*, 105–142.

Can, Y., & Grossberg, S. (2005). A laminar cortical model of stereopsis and 3d surface perception: Closuer and da vinci stereopsis. *Spatial Vis.*, *18*, 515–578.

Dietterich, T. G. (2000). Experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, *40*(2), 139–157.

Drucker, W. D. H., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Trans. on Neural Netw.*, *10*(5), 1048–1054.

Fawcett, T. (2003). ROC Graphs: Notes and practical considerations for data mining researchers. *Technical Report HPL-2003-4*. (HP Lab)

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition letters*, *27*(8), 861–874.

Ferris Research. (2005). The global economics impact of spam. *[Online], http://www.ferris.com/hidden-pages/reducing-the-50-billion-global-spam-bill/*.

Freund, Y. (2001). An adaptive version of the boost by majority algorithm. *Machine Learning*, *3*, 293–318.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proc. Int. Conf. Machine Learning*, pp. 148–165.

Freund, Y., & Schapire, R. E. (1997). Decision-theoretic generalization of on-line learning and application to boosting. *J. Computer and Syst. Sciences*, *55*(1), 119–139.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Grossberg, S. (1998). *Neural Substrates of Adaptively Timed Reinforcement, Recognition, and Motor Learning*, in Models of action: Mechanisms for adaptive behavior. In C. D. L. Wynne & J. E. R. Staddon (Eds.), (pp. 29–85). Hillsdale, NJ: Erlbaum Associates.

Grossberg, S. (1999). How does the cerebral cortex work? Learning, attention and grouping by the laminar circuits of visual cortex. *Spatial Vis.*, *12*, 163–185.

Grossberg, S. (2003). Adaptive resonance theory. *The Encyclopedia of Cognitive Science*. (Technical Report, CAS/CNS TR-2000-024).

Grossberg, S., & Howe, P. D. (2003). A laminar cortical model of stereopsis and three-dimensional surface perception. *Vis. Res.*, *43*(7), 801–829.

Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

He, H., & Chen, S. (2008). IMORL: Incremental multi-object recognition and localization. *IEEE Trans. Neural Networks*, *19*(10), 1727–1738.

He, H., Chen, S., Cao, Y., Desai, S., & Hohil, M. E. (2008). Multi-objects recognition for distributed intelligent sensor networks. *Proc. SPIE*, 6963, 69630R–69630R-10.

He, H., Chen, S., Cao, Y., & Starzyk, J. A. (2008). Incremental learning for machine intelligence. *Proc. Int. Conf. on Cognitive and Neural Systems*.

He, H., & Starzyk, J. A. (2007). Online dynamic value system for machine learning. *Lecture Notes in Computer Science*, *4491*, 441–448.

Ke, Y., & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. *Proc. IEEE Comput. Vis. Pattern Recognit*, 2, 506–513.

Lai, C. C., & Wu, C. H. (2007). Particle swarm optimization-aided feature selection for spam email classification. *Proc. Int. Conf. Innovative Computing, Information and Control*, pp. 165.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proc. Int. Conf. Conput. Vis.*, pp. 1150–1157.

Lowe, D. G. (2003). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, *20*.

Oza, N. C. (2003). Boosting with averaged weight vectors. In T. Windeatt & F. Roli (Eds.), *Int. Workshop Multiple Classifier Syst., Lecture Notes in Computer Science* (Vol. 2709, pp. 15–24). Springer.

Oza, N. C. (2004). AveBoost2: Boosting for noisy data. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *Int. Workshop on Multiple Classifier Syst., Lecture Notes in Computer Science* (Vol. 3077, pp. 31–40). Springer.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, *12*(1), 145–191.

Rumelhart, D. E., & McClelland, J. L. (1986a). *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1*. Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986b). *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 2*. Cambridge, MA: MIT Press.

Schapire, R. E., Freund, Y., Barlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, *26*(5), 1651–1686.

Scott, D. W. (1992). *Multivariate density estimation: Theory, practise, and visualization*. New York: Wiley-Interscience.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman & Hall/CRC.

Wang, D. L. (2005). The time dimension for scene analysis. *IEEE Trans. Neural Networks*, *16*(6), 1401–1426.

Whitworth, B., & Whitworth, E. (2004). Spam and the social technical gap. *IEEE Computer*, *37*(10), 38–45.

Yang, Y., & Elfayoumy, S. (2007). Anti-spam filtering using neural networks and bayesian classifiers. *Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pp. 272–278.

Zhang, H., & Li, D. (2007). Naïve bayes text classifier. *Proc. Int. Conf. Granular Computing*, pp. 708–708.

Zickler, S., & Veloso, M. (2006). Detection and localization of multiple objects. *Proc. Humanoids*, pp. 20–25.

第3章

不平衡学习

3.1 引言

近年来，不平衡学习问题已经引起了学术界和产业界的广泛关注。简单地说，不平衡学习问题主要关注在数据表示不充分和类分布扭曲变形时学习算法的性能(He&Garcia，2009)。由于不平衡数据集固有的复杂特点，学习这样的数据需要将大量的原始数据高效地转化为信息和知识表示的新理解、新原理、新算法、新工具。在实际领域中，不平衡学习是一个反复出现、影响广泛、值得深入探索的重要问题。近年几个主要研讨会、学术会议及专辑，如AAAI'00、ICML'03的不平衡数据学习研讨会(Chawla，Japkowicz & Kolcz，2003)和ACM SIGKDD Explorations'04(Chawla，Japkowicz & Kolcz，2004)也反映了对不平衡学习的浓厚兴趣。本章试图系统性地讨论不平衡问题的本质及当前此问题的最佳解决方法(He & Garcia，2009)。

3.2 不平衡学习的本质

在讨论解决不平衡学习问题的技术前，首先通过生物医学应用的一个具体例子理解不平衡学习的本质。考虑“乳腺透视图像数据集”，该数据集包含对不同患者检查得到的一系列乳腺透视图像，且已广泛用于解决不平衡学习问题(Chawla，Bowyer，Hall & Kegelmeyer，2002；Guo & Viktor，2004b；Woods等，1993)的算法分析。用二元分类分析这些图像时，用“阳性”标记“癌症”患者的透视图像，用“阴性”标记“健康”患者的透视图像。由经验可知，非癌症患者数量会远远超过癌症患者。事实确实如此，这个数据集包含了10 923个“阴性”(多数类)样本和260个“阳性”(少数类)样本。最理想的情况是分类器对少数类和多数类的预测准确度达到平衡(理想情况下为100%)。在实际中发现，分类器的准确度严重不平衡，例如，

对于多数类达到接近100%的准确度，而对于少数类的准确度仅有0～10%(Chawla等，2002；Woods等，1993)。假设一个分类器对乳腺透视图像集上的少数类的准确度达到10%，分析表明234个少数类的样本会被错分为多数类，这相当于有234个癌症患者被误诊断为健康。在医疗行业，这种结果对应的法律后果极为严重，甚于将一个非癌症患者诊断为癌症患者(Rao，Krishnan & Niculescu，2006)的影响。因此，对于这一领域的分类器，要求其在不严重影响多数类分类准确度的前提下对少数类要有足够高的准确度。这也进一步表明，传统的单一评价标准的评价方式，如总准确度或总误差率，无法提供足够的不平衡学习的信息。因此，对于不平衡数据学习的算法性能的最终评价，需要信息更加丰富的评价标准。这些问题将在本章的3.4节中讨论。除了在生物医学领域的应用外，在其他领域也有相似的结果，例如欺诈检测、网络入侵、溢油检测等(Kubat，Holte & Matwin，1998；Rao等，2006；Chan，Fan，Prodromidis & Stolfo，1999；Clifton，Damminda & Vincent，2004；Chan & Stolfo，1998)。

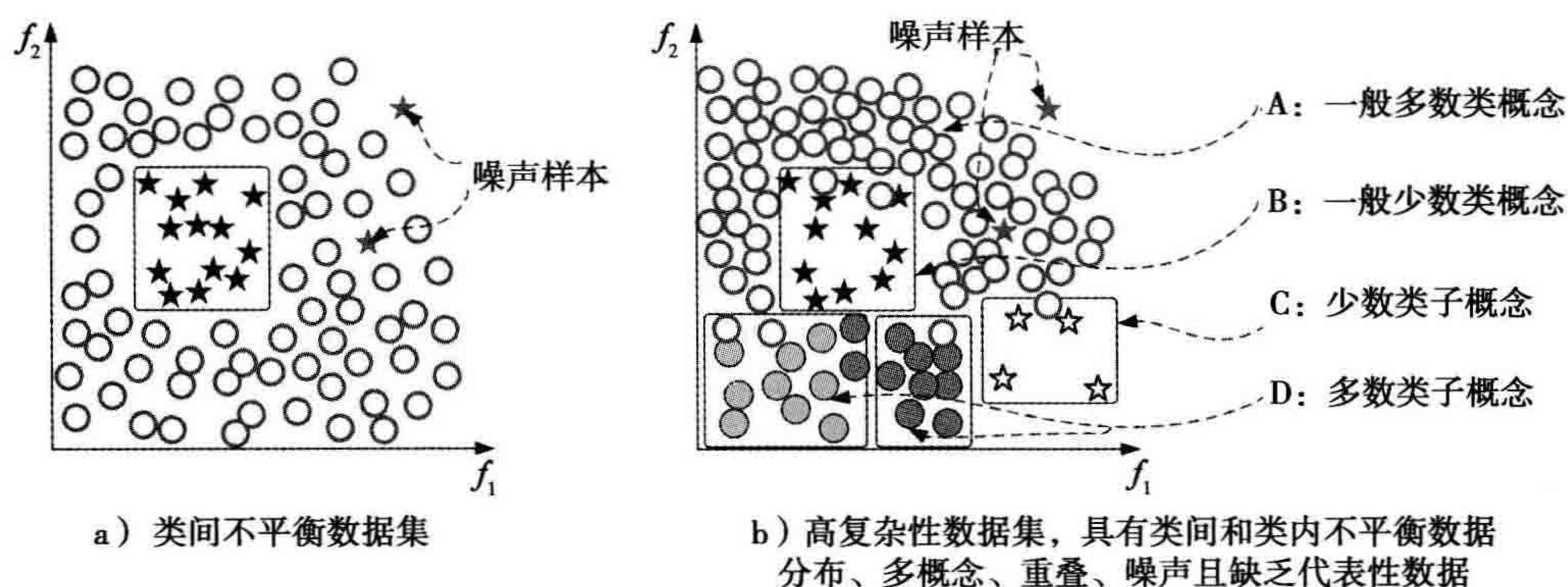
从技术上讲，任何表现出类分布不均衡的数据集都可以认为是不平衡的。但是，对不平衡数据的通常理解是指，数据集表现出显著的、在某些情况下甚至极端的不平衡。特别地，这种不平衡被称为类间不平衡；常见的类间不平衡可达到100∶1、1 000∶1和10 000∶1；在这些情况下，一个类的样本远远多于另一个类的样本(He & Shen，2007；Kubat等，1998；Pearson，Goney，& Shwaber，2003)。尽管这种描述似乎暗示所有的类间不平衡都是二元的(两分类)，但是多类数据中往往也存在着各类之间的不平衡(Sun，Kamel & Wang，2006；Abe，Zadrozny & Langford，2004；Chen，Lu & Kwok，2006；Zhou & Liu，2006；Liu & Zhou，2006b；Tan，Gilbert & Deville，2003)。

上述生物医学例子中的不平衡问题通常是固有的，即不平衡是数据空间本质的直接结果。然而，不平衡数据并不仅仅局限于数据空间本质的多样性。可变因素(如时间和存储)也会引起数据集的不平衡。这种形式的不平衡被认为是外在的，即不平衡不是数据空间本质的直接结果。外在不平衡与固有不平衡同样重要，因为经常发生这样的情况：外部不平衡数据集的数据空间可能是平衡的。例如，假设一个数据集是对平衡数据的连续数据流以特定时间间隔接收所得，如果在某个间隔内传输过程中有零星的间断，从而导致数据没有被传输，那么这会造成所得数据集不平衡。在这种情况下获得的数据集就是一个来自平衡数据空间的外部不平衡数据集(He & Garcia，2009)。

除了固有不平衡和外部不平衡，理解相对不平衡与稀有样例(绝对稀有)所导致

的不平衡之间的区别也很重要(Weiss，2004，2005)。考虑一个具有100 000个样例的乳腺透视图像数据集，类间不平衡比为100∶1，我们希望数据集中含有1000个少数类样例。显然，多数类占支配地位。假设通过测试更多的患者使得样本空间加倍，并进一步假设数据分布不变，即少数类包含2000个样例。显然，少数类仍然寡不敌众。然而，具有2000个样例的少数类相对自身来说已经不再稀少，但是相对于多数类来说仍然是稀少的。这是相对不平衡的一个典型例子。相对不平衡在实际应用中很常见，也是数据挖掘和数据工程的研究热点。一些研究表明，对于确定的相对不平衡数据集，少数类概念是在伴有少量干扰的情况下从不平衡数据中学习到的(Batista，Prati & Monard，2004；Japkowicz & Stephen，2002；Weiss & Provost，2003)。这些结果具有明显的启发性，因为它表明数据不平衡的程度不是唯一阻碍学习的因素。事实证明，数据集的复杂性是使分类性能变差的主要决定因素，而提高相对不平衡将增加数据集的复杂性。

数据集的复杂性是一个宽泛的术语，包括重叠、缺少代表性数据、小间隔等(He & Garcia，2009)。考虑一个简单的例子，以图3-1所示的数据分布为例，其中的星和圆圈分别代表少数类和多数类。通过观察，我们发现图3-1a和3-1b所示的数据分布均表现出相对不平衡。然而，图3-1a所示的数据分布类间没有重叠的样本，并且每个类只对应唯一一个概念，而图3-1b所示的数据分布有多个概念且相对应的样本存在严重的重叠。同样令人感兴趣的是图3-1b中的子概念C，由于它缺乏代表性数据，所以可能无法学习，我们将进一步进行讨论这种由小样本导致的具身不平衡。



a）类间不平衡数据集

b）高复杂性数据集，具有类间和类内不平衡数据分布、多概念、重叠、噪声且缺乏代表性数据

图3-1　不平衡数据学习问题

由小样本导致的不平衡是域中少数类样本非常有限时的典型问题，即目标概念稀有(He & Garcia，2009)。在这种情况下，忽略类间不平衡，缺乏代表性数据会使得算法学习非常困难(Weiss，2004)。此外，少数类概念可能额外包含样本有限

的子概念，这导致了不同程度的分类困难(Holte，Acker & Porter，2003；Quinlan，1986)。事实上，这是另一种形式的不平衡的结果，即类内不平衡，它关系到一个类自身与类内子概念的代表性数据的分布(Jo & Japkowicz，2004；Japkowicz 2003；Prati，Batista & Monard，2004)。这些观点再次强调了图 3-1 所示的例子。在图 3-1b 中，簇 B 代表少数类的概念，簇 C 代表少数类的子概念，簇 D 代表多数类的两个子概念，簇 A(不封闭)代表占优势的多数类概念。在这两类中，优势簇的样本数量明显多于其子概念簇中的样本数量，这使得数据空间表现出类间不平衡和类内不平衡。此外，如果想要完全删除簇 B 中的样本，数据空间将会产生一个均匀的、易于识别(簇 C)的少数类概念，但是其由于严重边缘化而无法被学习。

类内存在的不平衡与小间隔分离(small disjuncts)密切相关，小间隔分离对分类性能影响较大(Japkowicz & Stephen，2002；Jo & Japkowicz，2004；Japkowicz 2003；Prati 等，2004)。简单地说，小间隔分离问题可以理解如下：分类器尝试通过构造描述主概念的多个分离规则学习一个概念(Weiss，2004；Holte 等，2003；Quinlan，1986)。在概念同质的情况下，分类器通常构造大间隔分离规则，即该规则能覆盖大的属于主概念的簇。然而，在概念异质时，作为边缘化子概念的直接结果的小间隔分离规则，即覆盖属于主概念样本的小簇，将会出现(Weiss，2004；Holte 等，2003；Quinlan，1986)。更重要的是，由于分类器尝试学习多数类和少数类概念，所以小间隔分离问题不仅仅局限于少数类概念。相反，多数类的小间隔分离可能会产生于错分类的少数类样本或边缘化的子概念。但是，由于多数类的代表数据数量较大，这种情况并不常见。一种常见的情况是，噪声数据影响少数类的分离间隔。在这种情况下，对应于小间隔分离的簇的有效性成为了一个重要的问题，即这些样本代表的究竟是真实的子概念还是噪声数据。例如在图 3-1b 中，由于簇 A 中的少数噪声样本会导致分类器产生错误的、与簇 C 相当的簇，但簇 C 是一个真实的、被严重边缘化的子概念。

最后一个需要注意的问题是不平衡数据与小样本问题的结合(Raudys & Jain，1991；Caruana，2000)。在目前的数据分析和知识发现应用中，高维小样本数据已不可避免，如人脸识别和基因表达数据分析等。模式识别领域的小样本问题已被深入研究(Raudys & Jain，1991)，诸如主成分分析(prineipal component analysis，PCA)及其扩展等降维方法被广泛应用于此类问题(Yang，Dai & Yan，2008)。但是，当代表数据集的概念不平衡时，不平衡数据与小样本的结合引起了新的挑战(Caruana，2000)。这同时也提出了两个重要问题(Caruana，2000)。首先，由于样

本量很小，所有涉及绝对稀有和类内不平衡的方法都是可用的；其次，也是更重要的，出现不平衡问题时，学习算法常常无法将归纳规则推广到整个样本空间。这时，由于高维特征和有限样本难于结合，从而影响了学习效果。如果样本空间足够大，则可以为数据空间定义一组通用的(尽管复杂)规则。但是，当样本有限时，形成的规则过于特殊，这会导致过拟合。学习这样的数据集是一个比较新的、需要更多关注的研究主题，我们将在后续章节中继续讨论这一主题。

3.3 不平衡数据学习方法

3.2节对不平衡学习本质的讨论是现有研究这一挑战性问题的基础(He & Garcia，2009)。特别是，这些问题严重阻碍了标准学习算法的进展，也成为现有解决方案的研究重点。当在不平衡数据中应用标准学习算法时，由于少数类样本数量远远少于多数类且被边缘化，描述少数类概念的归纳规则往往弱于和少于多数类概念。

为了深入理解不平衡学习问题对标准学习算法的直接影响，我们以决策树学习算法为例进行分析。在这个例子中，在决策树的每个节点上，挖掘不平衡数据集分裂准则的不充分性(Japkowicz & Stephen，2002；Weiss & Provost，2003)，(Chawla，2003)。一般来说，决策树采用一个递归且自顶向下的贪婪搜索算法，该算法使用特征选择方法(如信息增益)来选择最佳特征，以作为树的每个节点的分裂准则。然后为每个对应于分裂特征的可能值创建叶子(Quinlan，1986；Mitchell，1997)。因此，训练集被依次划分成更小的子集，并最终用来构建关于类概念的不相交规则。最后，组合这些准则以使得最终分类器对所有类的误差率总和最小。在不平衡数据学习时，这一过程中的问题具有双重性。首先，对数据空间的连续分割，会导致对少数类样本的观察越来越少，从而使得描述少数类概念的叶子更少且置信估计不断降低。其次，由于分割引入的稀疏性，使得无法学习依赖于不同特征空间合并的概念。在这里，第一个问题与相对和绝对不平衡问题相关，而第二个问题与类间不平衡和高维度问题的相关性最大。这两种情况下，不平衡数据会影响决策树的分类性能。在下面的章节中，我们主要讨论克服不平衡数据对分类性能影响的方法。

为了清楚地描述，首先定义本节使用的一些符号。考虑一个给定的训练集 S，其包含 m 个样本(即$|S|=m$)，定义：$S=\{(x_i,y_i)\},i=1,\cdots,m$，其中 $x_i\in X$ 是 n 维特征空间 $X=\{f_1,f_2,\cdots,f_n\}$中的一个样本，$y_i\in Y\{1,\cdots,C\}$是样本 x_i 的类标签。

特别是，$C=2$ 代表两类分类问题。此外，定义子集 $S_{min}\subset S$、$S_{maj}\subset S$，其中 S_{min} 是集合 S 中的少数类样本集，S_{maj} 是集合 S 中的多数类样本集，因此 $S_{min}\cap S_{maj}=\{\Phi\}$，$S_{min}\cup S_{maj}=\{S\}$。最后，任何由对集合 S 的抽样形成的集合都被标记为 E，不相交的子集 E_{min} 和 E_{maj} 分别代表 E 的少数和多数类样本。

3.3.1 不平衡数据学习的抽样法

不平衡数据学习的抽样法是指通过一些机制修改不平衡数据集，使其变为平衡数据分布(He & Garcia，2009)。研究表明，对于一些基本分类器，在平衡数据集上的整体分类性能要高于其在不平衡数据集上的整体分类性能(Weiss & Provost，2001；Laurikkala，2001；Estabrooks，Jo & Japkowicz，2004)。然而，这并不意味着分类器不能通过不平衡数据集学习。相反，研究表明，分类器可以通过由抽样技术平衡的数据集进行学习(Batisa 等，2004；Japkowicz & Stphen，2002)。这一现象已直接把极少数情况下的问题及其相应的结果联系在一起，如 3.2 节所述。而且，对于大多数不平衡数据集，抽样技术的应用确实有助于提高分类器的分类准确度。

1. 随机过抽样和欠抽样

随机过抽样机制可以描述为在原始数据集中添加一个从少数类中选取的集合 E：对于一个从 S_{min} 中随机选择的少数类样本集合，可通过复制所选择的样本并将它们添加到原始数据集 S 中，从而扩充 S。通过这种方式，S_{min} 中的总样本数增加了 $|E|$，并且 $H(1)$ 中的类分布平衡相应地被调整。这种方法所提供的机制可以达到任意期望水平的不同程度的类分布平衡。过抽样方法易于理解和想象，因此我们不提供具体实例。

过抽样是向原始数据集中添加数据，而随机欠抽样是从原始数据集中删除数据。特别地，在 S_{maj} 中随机选择一个少数类样本集，并且从 S 中删除这些样本，从而得到 $|S|=|S_{min}|+|S_{maj}|-|E|$。因此，欠抽样是一种调整原始数据集 $H(1)$ 平衡程度的简单方法。

直观来看，过抽样和欠抽样方法看起来好像功能相似，因为它们既能改变原始数据集的大小，又能提供相同比例的数据分布平衡。然而，这种共性不只是表面的，这两种方法都有各自的问题，并潜在地影响学习(He & Garcia，2009；Holte 等，2003；Mease，Wymer & Buja，2007；Drummond & Holte，2003)。在欠抽样的情况下，这个问题比较明显(He & Garcia，2009)，从多数类中删除样本可能导致分类器丢失属于多数类的重要概念。至于过抽样，相关问题比较难理解：因为过抽样

简单地向原始数据中添加复制的数据，所以某些样本的过多样例会形成“僵局”，从而导致过拟合(Mease 等，2007)。特别是，分类器若对同一样本的多个拷贝产生多个分类准则，则会导致准则过于具体，从而会在过抽样中发生过拟合(He & Garcia，2009)。虽然这种情况下的训练准确度很高，但对于未知的测试数据，分类性能通常很糟糕(Holte 等，2003)。

2. **信息欠抽样**

文献 Liu、Wu 和 Zhou(2006)给出了两个比较成功的信息欠抽样算法：EasyEnsemble 算法和 BalanceCascade 算法。这两种算法的目的是克服传统随机欠抽样方法所存在的信息丢失的缺陷。EasyEnsemble 算法的实现非常简单：从多数类中独立抽样得到的若干个子集，并把每个子集分别与少数类数据相结合训练多个分类器。EasyEnsemble 可以看作一种无监督的学习算法，它采用替换的独立随机抽样来考察多数类数据。而 BalanceCascade 算法是一个监督学习算法，它通过系统地选择那些没有抽样过的样本作为多数类样本训练一个集成分类器。在集成过程中，对于第一个分类假设 $H(1)$，考虑多数类样本集 E，从而有 $|E|=|S_{min}|$，且 $N=\{E\cup S_{min}\}$，进而得到 $H(1)$。基于 $H(1)$的结果识别出所有属于 N 且被正确分类为属于 S_{maj} 的样本，并称它们的组合为集合 N_{maj}^{*}。因为已经得到了被训练过的 $H(1)$，因此可假设集合 N_{maj}^{*} 在 S_{maj} 中是冗余的。基于此，从 S_{maj} 中移除 N_{maj}^{*}，从而可得到新的多数类样本集 E，且有 $|E|=|S_{min}|$，$N=\{E\cup S_{min}\}$，基于集合 E 训练得到 $H(2)$。重复这个过程，在迭代中使用级联合并方法，直到满足终止条件，以得到最终的分类器(Liu 等，2006)。

另一个信息欠抽样的例子是使用 K-最近邻(KNN) 算法分类器实现欠抽样。根据给定的数据分布特点，Zhang and Mani(2003)中提出了 4 种 KNN 欠抽样算法，即 NearMiss-1、NearMiss-2、NearMiss-3 和“最远距离(Most distant)”方法。NearMiss-1 方法选择那些离少数类样本的平均距离最小的 3 个多数类样本；而 NearMiss-2 方法选择那些离少数类样本的平均距离最远的 3 个多数类样本；NearMiss-3 为每个少数类样本选择给定数量的、离其最近的多数类样本，以保证每个少数类样本被一些多数类样本环绕；最后，“最远距离”方法选择那些到 3 个最近少数类样本的平均距离最大的多数类样本。实验结果表明，对于不平衡数据学习，NearMiss-2 方法提供了具有竞争性的结果。

还有一些其他类型的信息欠抽样方法。例如，单边选择(OSS)方法(Kubat & Matwin，1997)选择多数类的代表性集合 E，并与所有少数类样本集合 S_{min} 结合，从而形成初始集合 N，$N=\{E\cup S_{min}\}$，使用数据清理技术使集合 N 进一步完善。

此方法将会在“数据清理抽样”小节中讨论，现在我们把注意力转向合成抽样方法。

3. 合成抽样的数据生成

关于合成抽样，合成少数类样本过抽样(SMOTE)技术是一种功能强大的方法，并在各种应用中取得了巨大成功(Chawla 等，2002)。SMOTE 算法基于已有的少数类样本的特征空间的相似性生成合成数据。特别地，对于子集 $S_{\min} \subset S$，其中每个样本 $x_i \in S_{\min}$的 K-最近邻(K 为指定的整数)被定义为 $S_{\min}$的 K 个元素，这些元素与当前样例 x_i 之间的欧氏距离在 n 维特征空间 X 中呈现出最小值。为了生成一个合成样本，随机选择 x_i 的一个 K-最近邻，计算其与 x_i 之间的差分向量，并乘以 1 个[0，1]区间内的随机数，最后加到 x_i 上：

$$x_{\text{new}} = x_i + (\hat{x}_i - x_i) \times \delta \tag{3-1}$$

其中，$x_i \in S_{\min}$为少数类样本，$\hat{x}_i$为对应于 x_i：$x_i \in S_{\min}$的 K-最近邻之一，且 $\delta \in [0, 1]$，是一个随机数。因此，根据式(3-1)产生的合成样本是在当前样例 x_i 与它的最近邻样例$\hat{x}_i$的特征向量的连接线段上选取的。

图 3-2 给出了一个 SMOTE 算法的实例。图 3-2a 表示一个典型的不平衡数据分布，其中，星和圆圈分别代表少数类和多数类。K-最近邻的数量设定为 $K=6$。图 3-2b 表示沿着 x_i 与$\hat{x}_i$ 之间的线段生成的样本(方形区域)。这些合成样本有助于打破由简单过抽样带来的僵局，并以显著提高学习的方式进一步扩大原始数据集。虽然这种方法拥有许多优点，但是 SMOTE 算法依然有自身的缺陷，包括过度泛化和不一致性(Wang & Japkowicz，2004)。我们将在下面的讨论中分析这些缺点。

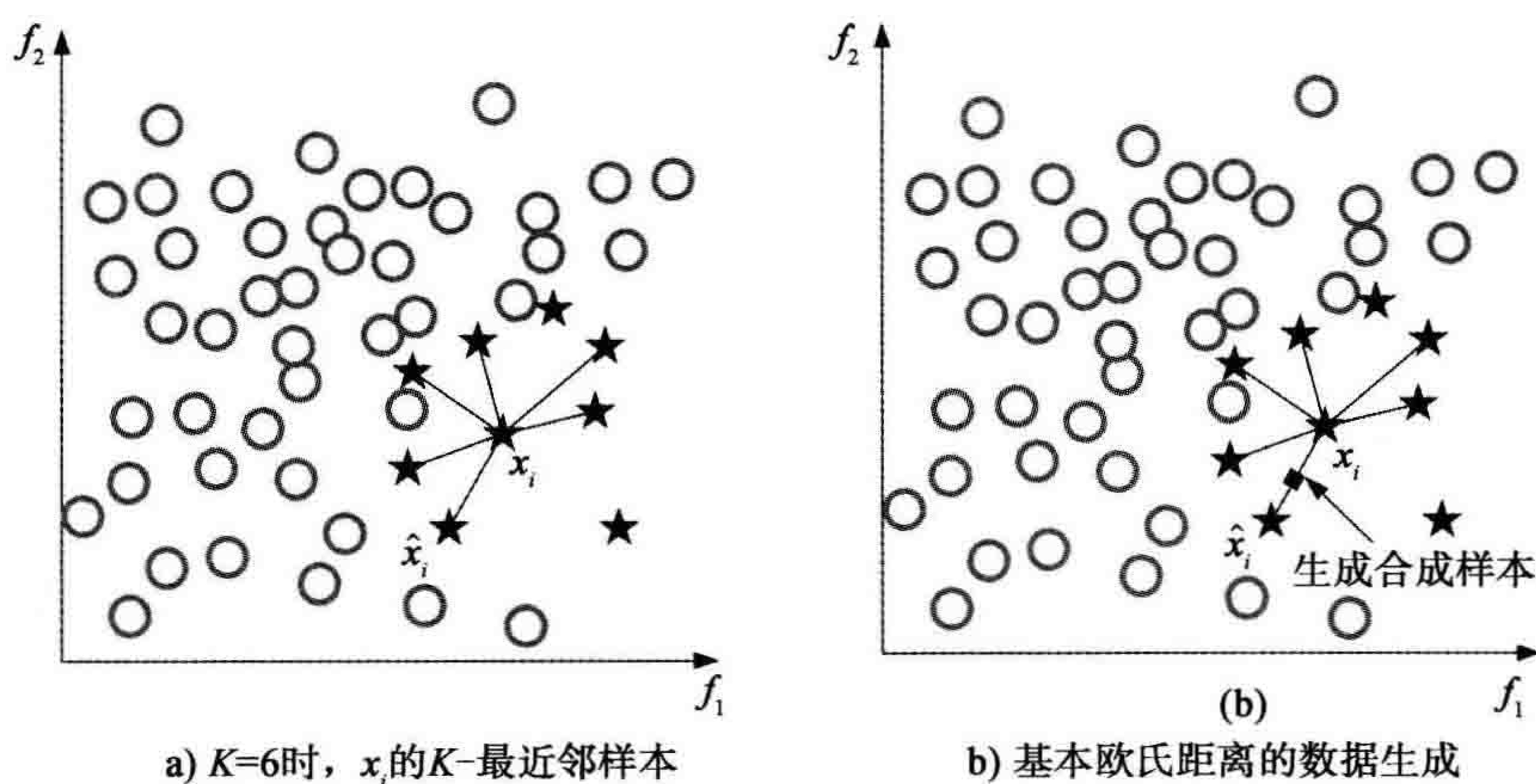


图 3-2 SMOTE 的数据生成

4. 自适应合成抽样

在 SMOTE 算法中，过度泛化问题在很大程度上是由于生成合成样本的方式所

导致的。特别地，对于每个原始少数类样本，如果 SMOTE 不考虑近邻样本，生成相同个数的合成数据样本，就会导致类之间发生重叠(Wang & Japkowicz，2004)。因此，人们提出了各种自适应抽样方法来克服这种局限性，代表性的工作包括 Borderline-SMOTE(Han，Wang & Mao，2005)和自适应合成抽样(ADASYN)(He，Bai，Gaicia，& Li，2008)算法。

特别令人感兴趣的是，这些自适应算法通常被用于识别少数类种子样本。Borderline-SMOTE 方法的实现过程如下：首先，确定每个 $x_i \in S_{\min}$ 的最近邻集合，称为 $S_{i:m-\mathrm{NN}}$，$S_{i:m-\mathrm{NN}} \subset S$；其次，对于每个 x_i，确定它的属于多数类的最近邻的数量，即 $|S_{i:m-\mathrm{NN}} \cap S_{\mathrm{maj}}|$；最后，选择满足下列条件的 x_i：

$$\frac{m}{2} \leqslant |S_{i:m-\mathrm{NN}} \cap S_{\mathrm{maj}}| < m \tag{3-2}$$

式(3-2)表明，只有那些多数类近邻元素多于少数类近邻元素的 x_i 才能被选定构成集合“DANGER”(Han 等，2005)。因此，集合 DANGER 中的样本代表边缘少数类样本(很可能被错误分类的样本)。然后集合 DANGER 被输入到 SMOTE 算法中，生成边缘附近的合成少数样本。图 3-3 展示了一个 Borderline-SMOTE 过程的例子。需要注意的是，如果 $|S_{i:m-\mathrm{NN}} \cap S_{\mathrm{maj}}| = m$，即 x_i 的所有 m 个最近邻都是多数类样本，如图 3-3 所示的样例 C，那么这样的 x_i 就会被当作是噪声，不会为它生成合成样本。对比图 3-3 和图 3-2，SMOTE 为每个少数类样例生成合成样例，而 Borderline-SMOTE 仅仅为那些“接近”边缘的少数类样例生成合成样例。

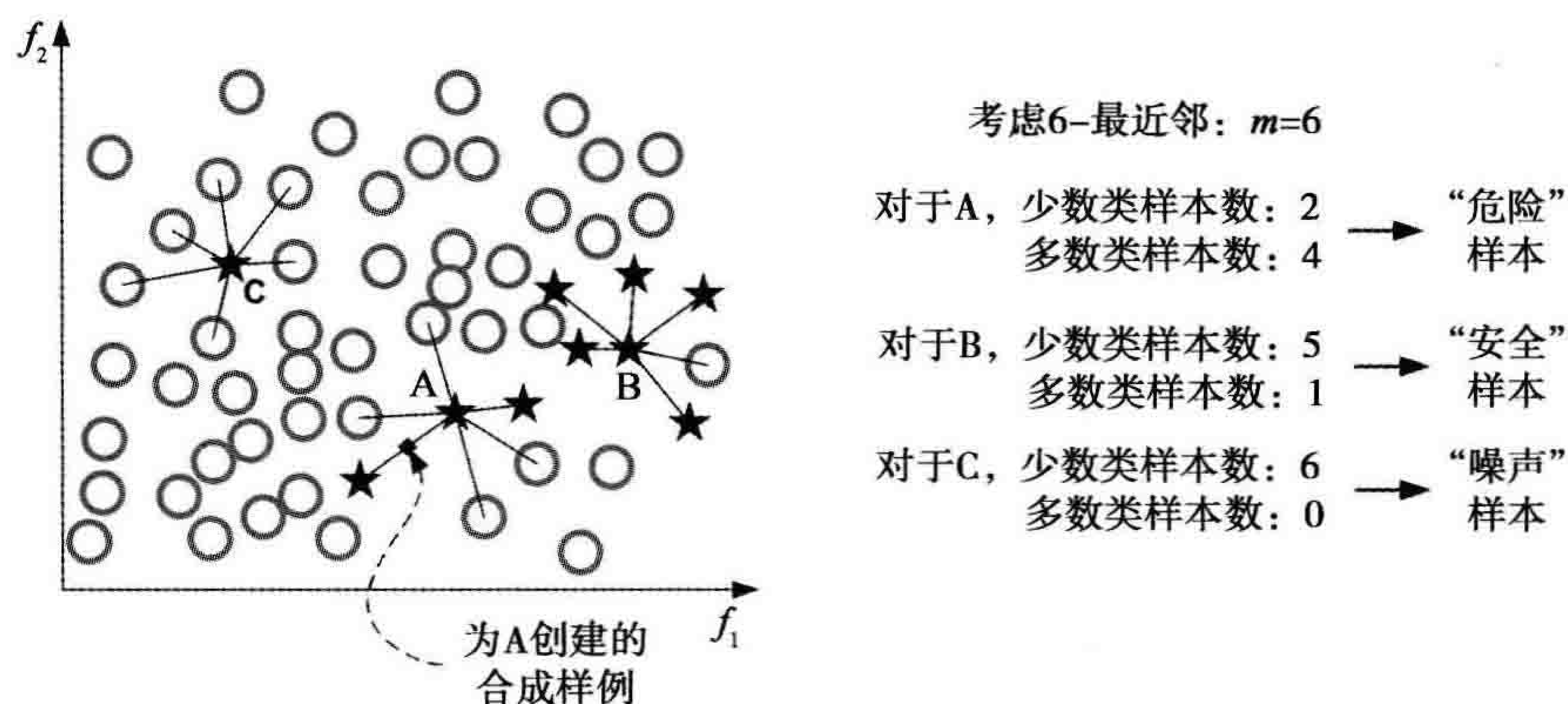


图 3-3 基于 Borderline-SMOTE 的数据生成实例

另一方面，ADASYN 利用合成方法，根据数据分布自适应地生成不同数量的合成数据(He 等，2008)。实现过程如下：首先，计算需要为全部少数类生成的合成数据样本的数量

$$G = (|S_{\mathrm{maj}}| - |S_{\min}|) \times \beta \tag{3-3}$$

其中，$\beta\in[0,1]$，是一个参数，用于指定合成数据生成后所期望的数据平衡水平。其次，对于每个样本 $x_i\in S_{\min}$，根据欧氏距离寻找 K-最近邻，并且计算比值 Γ_i，定义如下：

$$\Gamma_i=\frac{\frac{\Delta_i}{K}}{Z},\qquad i=1,\cdots,|S_{\min}| \tag{3-4}$$

其中，Δ_i是x_i 的属于S_{maj}的K-最近邻样本的数量，Z 是归一化常数，因此 Γ_i 是一个分布函数($\sum\Gamma_i=1$)。然后确定需要为每个 $x_i\in S_{\min}$生成的合成数据样本的数量：

$$g_i=\Gamma_i\times G \tag{3-5}$$

最后，对于每个 $x_i\in S_{\min}$，按照式(3-1)生成 g_i 个合成数据样本。ADASYN 算法的核心思想是以密度分布 Γ 为标准，自动确定需要为每个少数类样本生成的合成样本的数量，通过自适应地改变不同少数类样本的权重来抵消偏态分布。这样，ADASYN 可以根据数据分布适应性且系统性地生成合成数据样本，这点已经被在多种不同的数据集上(He 等，2008)的多个成功应用所证明。还应该注意到，SMOTE 中的最近邻计算仅考虑了少数类样本，而 ADASYN 和 Borderline-SMOTE 方法同时考虑了少数类样本和多数类样本(见图 3-3 和图 3-2)。

图 3-4 展示了 ADASYN 方法对于两类不平衡数据集、在不同系数 β 下的分类误差性能(He 等，2008)。训练数据集包含 50 个少数类样本和 200 个多数类样本，

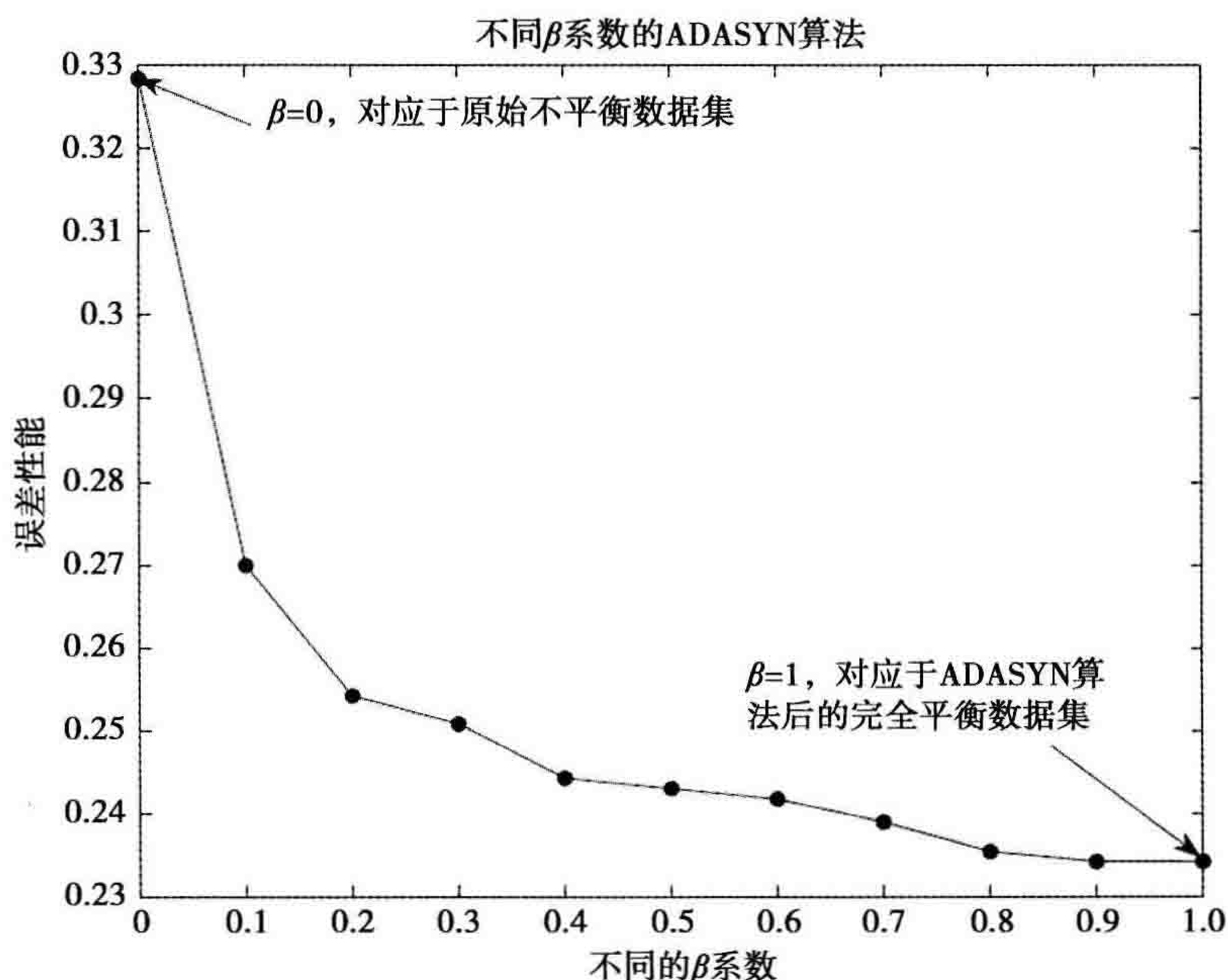


图 3-4 ADASYN 方法在两类不平衡数据集、不同系数 β 下的分类误差性能

测试数据集包含200个样本。所有的数据样本都是通过设置不同均值和协方差矩阵的多维高斯分布生成的。这些结果是100次运行的平均值，其中，决策树被用来作为基础分类器。在图3-4中，$\beta=0$ 对应于基于原始不平衡数据集的分类误差，$\beta=1$ 代表由ADASYN算法生成的充分平衡的数据集的分类误差。图3-4表明，ADASYN算法可以通过减少原始不平衡数据集的偏斜程度来提高分类性能，还表明，通过ADASYN算法，分类误差随着数据分布平衡水平的上升而降低。然而，这并不意味着，在一般情况下，算法的学习性能会随着数据分布平衡水平的上升而上升。实际上，现有的许多工作讨论了关于不平衡数据学习中的“最佳”和“理想”平衡率等级的问题(Weiss & Provost，2003；Estabrooks等，2004)。例如，在Estabrooks等(2004)中讨论的过抽样率和欠抽样率作为一种辅助手段取得了许多成功应用案例，实际上如何调试这些算法是一项非常具有挑战性的工作。为了缓解这一挑战，Estabrooks等人建议组合不同的抽样方法，这可能会有效解决此问题(Estabrooks等，2004)。对于一个固定大小的训练集，Weiss和Provost(2003)已经分析了训练数据的类分布(表示少数类样本的百分比)与分类器性能(在准确性和AUC(受试者工作特性[ROC]曲线下的面积，将在3.4.2节中详细讨论)方面)之间的关系。这项工作提供了关于“不同的训练数据类分布如何影响分类性能”和“哪种类分布提供了最好的分类器”的重要建议(Weiss & Provost，2003)。通过对26个数据集进行深入分析，发现如果将准确度作为性能评价标准，最好的类分布趋于接近自然产生的类分布。然而，如果将AUC作为评价标准，那么最好的类分布趋于接近平衡的类分布(Weiss & Provost，2003)。

5. 数据清理抽样

数据清理技术，如Tomek链接，已经被有效地用来删除由抽样方法引入的数据重叠。一般地，Tomek链接(Tomek，1976)可以被定义为一对相反类的最接近的最近邻。考虑一对样本：(x_i, x_j)，其中 $x_i \in S_{\min}$，$x_j \in S_{\mathrm{maj}}$，$d(x_i, x_j)$ 是 x_i 和 x_j 之间的距离，如果不存在这样的样本 x_k，满足 $d(x_i, x_k) < d(x_i, x_j)$ 或者 $d(x_j, x_k) < d(x_i, x_j)$，则称 (x_i, x_j) 为一个Tomek链接。这样，如果两个样本形成了Tomek链接，那么，要么其中一个样本是噪声，或者两者都接近边界。因此，在合成抽样后，可以利用Tomek链接来“清理”类间的多余重叠，删除所有的Tomek链接，直到所有的最近的最近邻对在同一个类中。通过消除重叠样本，可以通过建立定义明确的分类准则来改进分类性能。这个领域的一些具有代表性的工作包括单侧选择(OSS)方法(Kubat & Matwin 1997)；压缩最近邻规则和Tomek链接(CNN+

Tomek 链接)的集成方法(Batista 等，2004)；基于修改最近邻(ENN)规则的邻域清理规则(NCL)(Laurikkala，2001)，清除那些不同于三个之中的两个最近邻的样本；SMOTE 和 ENN 的集成算法(SMOTE＋ENN)以及 SMOTE 和 Tomek 链接的集成算法(SMOTE＋Tomek)(Batista 等，2004)。

图 3-5 显示了一个用 SMOTE 和 Tomek 清理重叠数据点(He & Garcia，2009)的典型过程。图 3-5a 显示了一个人造的不平衡数据集的原始数据集分布，注意少数类样本与多数类样本之间存在着固有重叠。图 3-5b 显示了通过 SMOTE 合成抽样后的数据集分布，可以看出，SMOTE 使得重叠增加。在图 3-5c 中用虚线框标出了 Tomek 链接。最后，图 3-5d 显示了实施清理后的数据集。可以看到，由该算法产生出了界限清晰的类簇，这有助于提高分类性能。而且，图 3-5 所示的想法也很重要，因为它引入了聚类的思想。在接下来的基于聚类的抽样算法的讨论中，将进一步研究聚类。

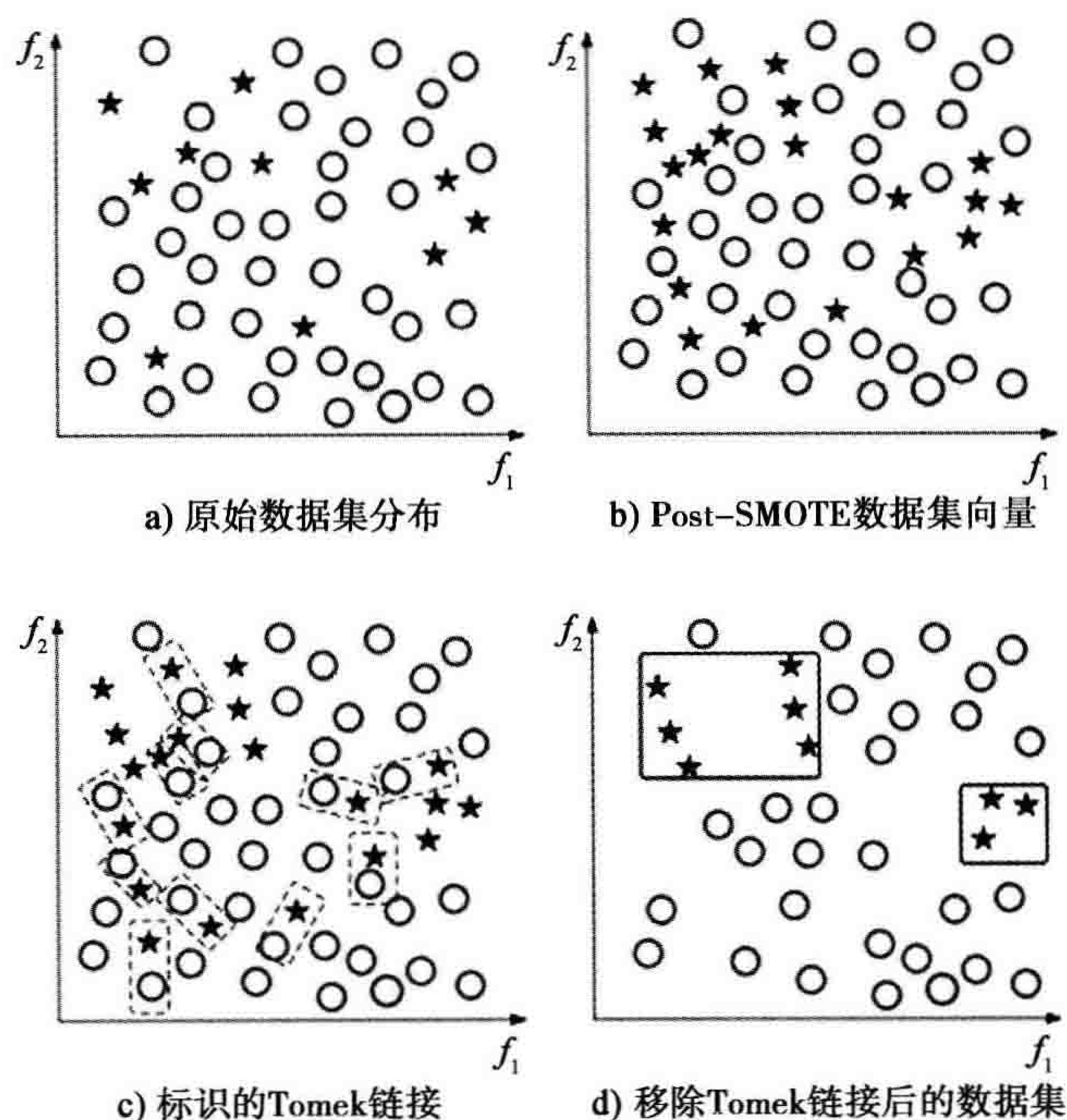


图 3-5　数据清理抽样

6. 基于聚类的抽样方法

基于聚类的抽样算法的吸引力主要在于它具有简单算法和合成抽样算法无法提供的灵活性，也因此被设计为针对特定问题的抽样方法。例如，Jo 和 Japkowicz (2004)所提出的基于聚类的过抽样(CBO)算法可有效地处理同时存在类间不平衡与

类内不平衡的不平衡数据问题。

CBO算法利用了K-均值聚类技术。从每个簇(对两类均做这样的处理)中随机选择K个样本，将这些样本的均值作为聚类中心。然后计算其余训练样本到聚类中心的欧氏距离，并将每个训练样本分配到离它最近的簇中。最后，更新所有簇的均值，一直重复此步骤，直到所有的样本都用尽(即，对于每个样本，只有一个聚类均值在本质上被更新)。

图3-6给出了具体的步骤(He & Garcia，2009)。图3-6a显示了原始数据分布，其中，多数类有3个簇A、B、C($m_{maj}=3$)，每个簇分别有20、10、8个样本。少数类有两个簇D和E($m_{min}=2$)，每个簇分别有8和5个样本。图3-6b显示了每个簇中3个随机样本的聚类均值(三角形所示)，即$k=3$。图3-6b还显示了5个所引入的独立样本x_1、x_2、x_3、x_4和x_5的距离向量。图3-6c显示了由于引入5个样本后更新的聚类均值和聚类边界。当用尽所有样本时，CBO算法利用过抽样方法膨胀除了最大簇以外的所有多数类簇，以便使所有的多数类簇与最大簇具有相同的大小(即，簇B和C每个都有20个样本)。过抽样后的多数类样本总数被记为N_{CBO}，则$N_{CBO}=|S_{maj}|+|E_{maj}|$(如，当前例子中$N_{CBO}=60$)。然后对多数类进行过抽样，以便使每个簇包含$\frac{N_{CBO}}{m_{min}}$个样本(即，每个少数类簇D和E在经过过抽样后将有60/2=30个样本)。图3-6d显示了应用CBO方法后的最终数据集。与图3-6a对比，可以看出，最终数据集对稀有概念具有较强的表达。请注意，可以把不同的过抽样方法集成到CBO算法中。例如，在“随机过抽样和欠抽样”小节讨论的Jo和Japowicz(2004)采用的随机过抽样方法，虽然图3-6中的实例揭示了不平衡数据学习的本质，即针对不平衡数据集，将类内和类间不平衡串联处理是一种有效策略。

7. 抽样与自举的集成

机器学习领域已开始了集成抽样策略与集成学习的研究。例如，使用Adaboost.M2方法的SMOTE-Boost。特别地，SMOTE-Boost方法把合成抽样引入到每一次的自举迭代中，在不同数据抽样的基础上逐次集成分类器，最终由投票选出分类器，这种分类器对少数类会有一个加宽的且定义明确的决策区域。

另一个集成方法是DataBoot-IM(Guo & viktor，2004b)方法，把Guo和viktor(2004a)中所描述的数据生成技术与AdaBoost.M1相结合，在不牺牲多数类准确度的情况下，对少数类实现了较高的预测准确度。简单地说，DataBoost-IM根据类间的困难学习样本的比例生成合成样本。具体地，对于数据集S和对应的子集$S_{min}\subset S$、$S_{maj}\subset S$，加权分布D_t代表学习每个样本$x_i\in S$的相对困难度。根据各自的权重降

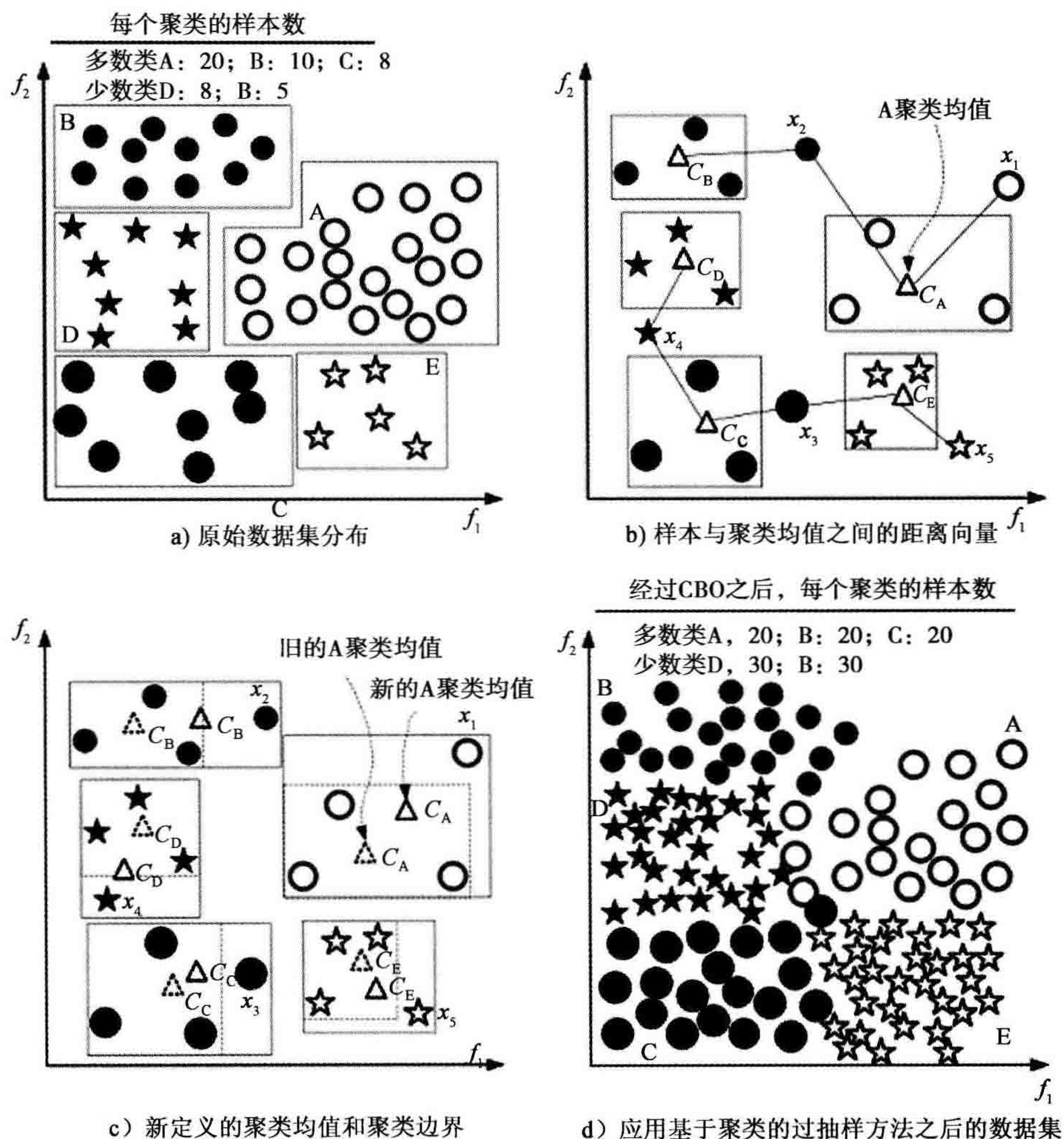


图 3-6　基于聚类的过抽样

序排列 x_i，然后选择顶部$|S|\times\text{error}(t)$个样本填充集合 E 且 $E\subset S$，其中 $\text{error}(t)$ 是当前学习的分类器的错误率。因此，E 是从两类中搜集的困难学习样本的集合，所以存在子集 $E_{\min}\subset E$ 和 $E_{\text{maj}}\subset E$。此外，由于学习少数类样本通常比学习多数类样本更加困难，所以可以预测$|E_{\text{maj}}|\leqslant|E_{\min}|$。

一旦确定了困难样本，那么 DataBoost-IM 算法可利用下列双重过程继续创建合成样本：首先，识别形成合成样本的集合 E 的“种子”，然后基于这些样本生成合成数据。种子识别过程是以 E 和 S 中类表示的比率为基础的。多数类种子的数量 M_{L} 定义为 $M_{\text{L}}=\min\left(\frac{|S_{\text{maj}}|}{|S_{\min}|},\ |E_{\text{maj}}|\right)$，少数类种子的数量 M_{S} 定义为 $M_{\text{S}}=\min\left(\frac{|S_{\text{maj}}|\times M_{\text{L}}}{|S_{\min}|},\ |E_{\min}|\right)$。继续合成数据集 E_{syn} 的两个子集 $E_{\text{smin}}\subset E_{\text{syn}}$ 和 $E_{\text{smaj}}\subset$

E_{syn}，使得$|E_{smin}|=M_S\times|S_{min}|$，$|E_{smaj}|=M_L\times|S_{maj}|$。然后集合$S$被$E_{syn}$扩大，少数类样本数量增加，数据分布更平衡。最终，在考虑新添加的合成样本的情况下更新加权分布D_t。

为了集成ADASYN和自举技术的优点，最近一项研究提出了一种基于自举的排序少数类过抽样方法，即RAMOBoost方法(Chen，He & Garcia，2010)。简单地说，RAMOBoost方法在每个学习迭代过程中，根据基于底层数据分布的抽样概率分布，自适应地排列少数类样本，而且通过使用一个分类器评估过程自适应地把决策边界移向难以学习的少数类和多数类样本。为了减缓数据分布的影响，RAMOBoost的双重目标可通过两个方面来实现：首先，在RAMOBoos中嵌入一个自适应权重调节过程，从而把决策边界转向少数类和多数类中的难以学习的样本；然后，用排序抽样概率分布生成合成估计来平衡偏态分布。通过这种方式，RAMOBoost评估每个少数类样本的潜在学习贡献，并以此为根据相应地决定其抽样权重。具体而言，这是通过计算任意单个少数类样本与最近邻集合之间的距离来决定它对学习过程的贡献有多大。

为了深入理解这种方法，这里给出一个例子，该数据集包含2000个多数类样本和100个少数类样本，在这个数据集上比较RAMOBoost(Chen等，2010)、SMOTE(Chawla等，2012)和ADASYN(He等，2008)的数据生成机制。图3-7a显示了原始的不平衡数据分布，图3-7b～d分别显示了应用SMOTE之后的数据分布、应用ADASYN之后的数据分布和应用RAMOBoost之后的数据分布。在这些图中，"×""+"和"•"分别代表原始多数类数据、原始少数类数据，以及生成的合成数据。而且，每一个图还显示了用于性能估计的分类混淆矩阵(瞬时计时)。对比每幅图的混淆矩阵可以看出，用RAMOBoost方法可以提高分类性能。特别地，随着原始数据集的变化，SMOTE的真阴性(TN)计数提高了，从1992增加到1993，而RAMOBoost是从1992增加到1998。这是因为，在SMOTE中，对于每个少数类样本会生成相同数量的样本，而在RAMOBoost中，数据合成过程是按照数据分布自适应的。

从图3-7c可以看出，ADASYN似乎非常激进地从边缘学习，因为它生成的合成数据样本非常接近决策边缘。这可能对学习性能产生两方面的影响：提高少数类数据的分类准确性，因为接近边缘的少数类数据分布可以被很好地表示(因此提高了查全率(recall)(参见3.4节中详细讨论的不平衡数据学习的评价标准))；降低多数类的分类性能，转而恶化整体分类性能。从图3-7c可以观察到，尽管ADASYN技术对少数类样本的分类准确度是这3种方法中最好的(真阳性(TP)=100，查全率=1)，但

是 ADASYN 的真阴性计数也明显减小了(所有情况下的最低值为 TN＝1986)。为此，RAMOBoost 可以同时利用 SMOTE 和 ADASYN 的优势来提高整体学习性能(Chen 等，2010)。

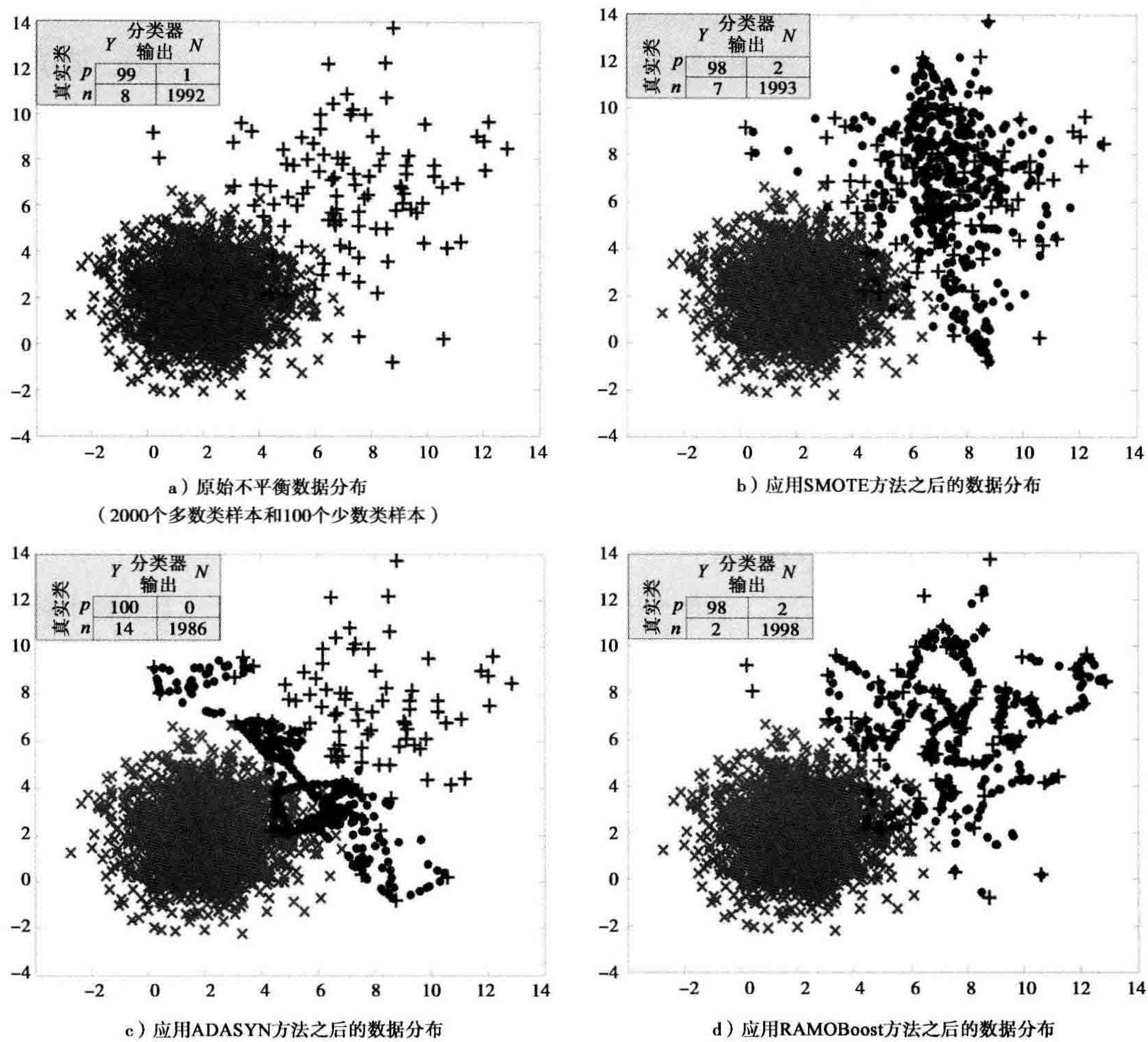


图 3-7 不同的合成数据生成机制的比较

证据表明，合成抽样方法可以有效地处理不平衡数据学习问题，然而，到目前为止所讨论的数据生成方法都是复杂的并且计算代价高。考虑“随机过抽样和欠抽样”小节所讨论的随机过抽样中的“僵局”问题，Mease 等人(2007)提出了一种更简单的方法来打破这些“僵局”：不是用计算方法生成新数据，而是在随机过抽样获得的复制数据中引入扰动(“抖动”)，从而打破这种僵局。由此产生的算法——过/欠抽样抖动(JOUS-Boost)，在每一次自举迭代过程中为少数类样本引进了独立同分布(iid)噪声，过抽样这些少数类样本的合成抽样副本，也有益于提高集成度，从而改善算

法的性能。实证研究已经证明了这种方法非常有效，这也表明合成过程是非常成功的，不增加运行成本。

3.3.2 不平衡数据学习的代价敏感方法

抽样方法试图调整数据分布中各类样本的比例来平衡分布，而代价敏感学习方法则考虑对样本错误分类的代价(Elkan，2001；Ting，2002)。代价敏感学习不是通过不同的抽样策略来创建平衡的数据分布，而是采用不同的代价矩阵来解决不平衡数据的学习问题，代价矩阵描述了特定数据样本错误分类的代价。研究表明，代价敏感学习与不平衡数据学习之间存在着紧密的联系，因此，代价敏感方法的理论基础和算法可以很自然地应用于不平衡数据学习问题(Chawla 等，2004；Weiss，2004；Maloof，2003)。而且，各种实证研究表明，在一些应用领域，包括某些不平衡学习领域(Liu & Zhou，2006a，2006b；McCarthy，Zabar & Weiss，2005)，代价敏感学习要优于抽样方法。因此，代价敏感技术为不平衡数据学习领域的抽样方法提供了一种可行的替代方法(He & Garcia，2009)。

1. 代价敏感学习框架

代价敏感学习方法的基础是代价矩阵的概念。代价矩阵可以看作是以数值形式表示将样本从一类错分到另一类的惩罚进行定量表示。例如，在一个二元分类方案中，定义 $C(\mathrm{Min},\mathrm{Maj})$为把一个多数类样本错分为少数类样本的代价，$C(\mathrm{Maj},\mathrm{Min})$代表相反情况下的代价。通常情况下，$C(\mathrm{Min},\mathrm{Maj})$比 $C(\mathrm{Maj},\mathrm{Min})$低，即 $C(\mathrm{Maj},\mathrm{Min})>C(\mathrm{Min},\mathrm{Maj})$。代价敏感学习的目标是建立一个能够减少训练数据集总代价的分类器，这通常是贝叶斯条件风险。这些概念很容易通过 $C(i,j)$扩展为多类数据，$C(i,j)$代表当真实类为 j 时预测类 i 的代价，其中 $i,j\in Y=\{1,\cdots,C\}$。图3-8显示了多类问题的代价矩阵。这种情况下，条件风险定义为 $R(i|x)=\sum_j P(j|x)C(i,j)$，其中 $P(j|x)$代表每个类 j 对于给定样本 x 的概率(Elkan，2001；Domingos，1999)。

真实类 j

预测类 i	1	2	…	k
1	$C(1,1)$	$C(1,2)$	…	$C(1,k)$
2	$C(2,1)$	…	…	⋮
⋮	⋮	…	…	⋮
	$C(k,1)$	…	…	$C(k,k)$

图3-8 多类代价矩阵

实现代价敏感学习的方法很多，主要技术可以归为以下三类(He & Garcia，2009)。第一类技术把错误分类代价作为数据空间的权重应用于数据集，这类技术本质上是代价敏感自举抽样方法，用错误分类代价来选择最佳训练分布。第二类方法是将代价最小化技术应用到集成方法的组合方案中，包括各种 Meta 技术，其中，将标准学习算法与集成方法相结合来开发代价敏感分类器。这两类方法都具有丰富的理论基础，证明了代价敏感数据空间加权方法是建立在平移定理上的(Zadrozny，Langford & Abe，2003)，代价敏感 Meta 技术是建立在 Metacost 框架上的(Domingos，1999)。实际上，大多数现有的研究往往集成了 Metacost 架构、数据空间加权方法、自适应自举方法，以实现更好的分类结果。为此，在下面的章节中将这两类算法当作一类。最后一类技术直接把代价敏感函数或特性合并入分类模式中，以使代价敏感框架“适合”这些分类器。由于许多这样的技术都是针对特定范例的，因此不存在此类代价敏感学习的统一框架，但是在很多情况下，一个范例的工作方案往往可以抽象到其他范例的工作中。

2. 自适应自举代价敏感数据空间加权

受 AdaBoost 算法(Freund & Schcpire，1996，2002)的启发，人们提出了不平衡数据学习的代价敏感自举方法。Sun、Kamel、Wong 和 Wang(2007)等通过在 AdaBoost 的权重更新策略中引入代价项的方式提出了 3 个代价敏感自举方法：Adac1、AdaC2 和 AdaC3。Adaboost. M1 方法的主要思想是对训练数据集的分布函数进行迭代更新，每次迭代 $t(=1, \cdots, T)$，其中 T 是预设的总迭代次数，分布函数 D_t 被序贯更新，并被用来训练新的分类器：

$$D_{t+1}(i) = D_t(i)\exp(-\alpha_t h_t(x_i) y_i)/Z_t \tag{3-6}$$

其中，$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$是权重更新系数；$h_t(x_i)$是分类器 h_t 对样本 x_i 的预测输出，ε_t 是分类器 h_t 在训练数据 $\varepsilon_t = \sum\limits_{i:h_t(x_i)\neq y_i} D_t(i)$ 上的误差；Z_t 是归一化因子，它使得 D_{t+1} 是一个分布函数，即 $\sum\limits_{i=1}^{m} D_{t+1}(i) = 1$。基于这些描述，代价因子可以用于指数内、指数外及指数内外，解析式分别为

$$D_{t+1}(i) = D_t(i)\exp(-\alpha_t C_i h_t(x_i) y_i)/Z_t \tag{3-7}$$

$$D_{t+1}(i) = C_i D_t(i)\exp(-\alpha_t h_t(x_i) y_i)/Z_t \tag{3-8}$$

$$D_{t+1}(i) = C_i D_t(i)\exp(-\alpha_t C_i h_t(x_i) y_i)/Z_t \tag{3-9}$$

式(3-7)～式(3-9)分别对应于 AdaC1、AdaC2 和 AdaC3 方法。这里，代价项 C_i 是与每个 x_i 相关的代价，C_i 的较高值对应于具有较高错误分类代价的样本。实质上，

这些算法使每次迭代中高代价样本被抽中的概率增大，训练数据集中高代价样本数量增加，从而使得到的分类器的归纳性能更有针对性。一般来说，将代价因子包括在 Adaboost 的加权方案中，可使决策偏向少数类概念，使每个分类器使用更多相关数据样本，分类结果更加鲁棒。

另一个类似的代价敏感自举算法是 AdaCost（Fan，Stolfo，Zhang & Chan，1999）。AdaCost 类似于 AdaC1，把代价敏感引入到 Adaboost 的权重更新公式的指数内。然而，AdaCost 没有直接使用代价项，而是使用一个代价调节函数，这极大地增加了高代价错误分类的权重，适当地减少了正确分类的高代价样本的权重。这个解析式表示为

$$D_{t+1}(i) = D_t(i)\exp(-\alpha_t h_t(x_i) y_i \beta_i)/Z_t \tag{3-10}$$

代价调节函数 β_i 定义为 $\beta_i=\beta(\text{sign}(y_i, h_t(x_i)), C_i)$，其中 $\text{sign}(y_i, h_t(x_i))$ 为“＋”表示正确分类，为“－”表示错误分类。为了清楚地描述，当 $\text{sign}(y_i, h_t(x_i))=1$ 时，可以使用 β_+；当 $\text{sign}(y_i, h_t(x_i))=-1$ 时，使用 β_-。该方法还注重在强调样本的重要性的同时允许充分的灵活性。例如，Fan 等人（1999）表示 $\beta_+=-0.5C_i+0.5$ 和 $\beta_-=0.5C_i+0.5$ 在大多数的应用中具有较好的结果，但是这些系数可以根据需要调整。Sun 等人（2007）在 4 个不平衡数据集上针对 AdaC1、AdaC2、AdaC3、AdaCost 以及另外两个相似的算法 CSB1、CSB2（Ting，2000），使用决策树和一个准则关联系统作为基础分类器以进行比较分析，实验结果表明，用 F-度量（参见 3.4.1 节）作为评价标准，在各种情况下，增强的集成分类器优于单一的基础分类器，在几乎所有的情况下，代价敏感增强的集成比普通的自举表现出更好的性能。

虽然这些代价敏感算法可以明显提高分类性能，但想当然地认为代价矩阵及其相关的代价项存在。然而，多数情况下，错误分类代价的确定性描述并不清楚，即只有一个非正式的描述，如对阳性类的错误分类比对阴性类的错误分类代价更高（Maloof，2003）。而且，确定一个给定域的代价表示可能非常具有挑战性，甚至在某些情况下是不可能的（Maloof，Langley，Sage & Binford，1997）。因此，本节讨论的技术并不适用于这些情况，必须寻求建立其他的解决方案，这是前面提到的代价敏感关联技术的主要动机。在下面的章节中，我们概述了基于这些方法的两种常用的学习模型，即决策树和神经网络。

3. 代价敏感决策树

对于决策树，可以采用 3 种形式与代价敏感关联：①代价敏感调整可以应用于决策阈值；②每个节点的分裂准则要考虑代价敏感因素；③代价敏感的剪枝方案可

以应用到树。

针对未知错误分类代价的不平衡数据，Maloof(2003)提出了一种决策树阈值移动方案。Breiman、Friedman、Olshen 和 Stone(1984)建立了每类错误分类代价、训练样本的分布以及决策阈值的设置之间的关系。然而，Maloof(2003)指出，这些关系的精确定义太具体，会导致基于这些关系的阈值选择方法难以实现。因此，他提出的算法不依靠训练数据分布或准确的错误分类代价，而是使用 ROC 评估过程(参见 3.4.2 节)。从 ROC 曲线可以看出决策阈值从阳性类的总错误分类代价的最大点移动到阴性类的总错误分类代价的最大点时算法的性能范围。在 ROC 曲线上最具优势点的决策阈值将作为最终决策阈值。

当在分裂准则中考虑代价敏感时，首要任务是拟合一个对不平等代价不敏感的混杂度函数(impurity function)。例如，传统意义上，准确度作为决策树的混杂度函数，用于在每个节点以最小误差选择分裂。然而，这个指标对样本分布的变化敏感(参见 3.4.1 节)，因此对不平等代价是天生敏感的。在 Drummond 和 Holte(2000)中，有 3 个特殊的混杂度函数 Gini、Entrop 和 DKM，与准确度/误差率基准相比，它们提高了代价不敏感性。而且，实证研究还证明了用 DKM 函数一般会生产较小的未剪枝决策树，与 Gini 和 Entropy 相比，准确度较低。随后在 Elkan(2001)中建立了解释这些实验结果的理论基础，分析了分裂准则的选择对决策树生长的影响。

最后一种适合代价敏感决策树的是决策树的剪枝。剪枝对决策树是有利的，因为通过移除那些类概率估计低于特定阈值的叶子可以提高算法的泛化性。然而，在不平衡数据存在的情况下，剪枝过程倾向于移除描述少数类概念的叶子。已经证明，虽然不平衡数据导致的决策树剪枝可能降低决策树的性能，但是在这种情况下，用未剪枝的决策树也不能提高性能(Japkowicz & Stephen，2002)。因此，应该注重提高每个节点的类概率估计来开发更具代表性的决策树结构，使剪枝具有积极的影响。这方面已经出现了一些代表性的工作，包括概率估计拉普拉斯平滑法和拉普拉斯剪枝技术(Elkan，2001)。

4. 代价敏感神经网络

代价敏感神经网络已经在不平衡学习领域得到了广泛研究。神经网络通常由一组密集的相互连接的简单神经元组成。神经网络分类器的大多数实际应用都包含一个多层结构，例如，常见的多层感知器(MLP)模型(Haykin，1999)，并且用反向传播算法与梯度下降规则相结合来促进学习。具体地，假设误差函数为

$$E(\omega)=\frac{1}{2}\sum(t_k-o_k)^2 \tag{3-11}$$

其中，ω代表一组需要训练的权值；t_k、o_k分别是神经元k的目标值、网络输出值。梯度下降规则的目的是寻找最陡下降，以修改每次迭代的权值：

$$\Delta\omega_n = -\eta\nabla_\omega E(\omega_n) \tag{3-12}$$

其中，η是指定的神经网络学习率；∇_ω表示相对于权重ω的梯度算子。而且，输出的概率估计可以通过规范化所有神经元的输出值来定义。

这个框架可以用4种方式将代价敏感引入到神经网络中(Kubar & Kononenko，1998)：①代价敏感调整可以用于概率估计；②可以使神经网络输出(即，每个o_k)具有代价敏感性；③代价敏感调整可以用于学习率η；④最小化误差函数可以用于说明预期代价。

对于概率估计，Kular和Kononenko(1998)把代价因子集成到分类测试阶段以自适应地修改神经网络输出的概率估计。这有利于保持神经网络的原始结构(和输出)，同时，通过考虑代价，加强了对高代价类的原始估计。Kular和Kononenko(1998)中的实证研究结果表明，该技术提高了原始神经网络的性能，但是改进并不大。然而我们注意到，通过在一个给定的数据集上使用交叉验证技术将这个估计应用到集成方法中，可以更加显著地提高性能；Liu和Zhou(2006b)使用了类似的方法，但是采用了稍微不同的估计。

第二种神经网络代价敏感关联技术直接改变神经网络的输出。在Kubar和Kononenko(1998)中，神经网络的输出在训练阶段被改变为偏向代价高的类。这种方法的实证研究结果表明，与估计数据集上的最小预期代价相比，这种方法的分类性能得到了平均提高。我们推测集成方法应该可以缓解这一问题，但据我们所知，到目前为止，这样的实验尚未进行。

学习率η也会影响权重的调整(参见式(3-12))。因此，代价敏感因子可以用学习率去改变对权重的影响——代价高的样本对权重变化的影响较大。这种方法的主要思想是，在学习期间，通过有效降低每个高代价样本的学习率，把更多的注意力放在高代价样本上。这也表明，低代价样本在训练期间达到了平衡。有关实验结果表明，这项技术对于训练神经网络显著改善基础分类器是非常有效的(Kubar & Kononenko，1998)。

代价敏感神经网络的最终改进形式是用期望的代价最小化函数代替式(3-11)中的误差最小化函数。这种形式的代价敏感被证明是本节所讨论的最主要的方法(Kubar & Kononenko，1998)。这也正符合了反向传播方法论、理论基础，该理论是建立在误差最小化和代价最小化分类器之间的传递性上的。

虽然我们只讨论了决策树和神经网络，其他学习模型中也存在许多代价敏感关联技术。例如，大量研究工作都集中在代价敏感贝叶斯分类器(Domingos & Pazzani，1996；Webb & Pazzani，1998；Kohavi & Wolpert，1996；Gama，2003)上，一些研究工作还将代价函数与支持向量机相结合(Fumera & Roli，2002；Platt，1999；Kwok，2003)。有兴趣的读者可以参阅相关文献进行更广泛的了解。

3.3.3 基于核的不平衡数据学习方法

当前，在不平衡数据学习研究中，虽然抽样方法和代价敏感方法占据了主导地位，但是许多其他方法也得到了学术界的追捧。在这一节中，我们简要回顾基于核的学习方法。由于基于核的学习方法为当今许多数据工程应用提供了先进的技术，借助基于核的方法来理解不平衡数据学习自然吸引了越来越多的关注(He & Garcia，2009)。

1. 基于核方法的学习框架

基于核方法的学习原理主要集中在统计学习理论和 Vapnik-Chervonenkis(VC)维度(Vapnik，1995)理论。基于核方法的典型学习模型支持向量机(SVM)，当应用不平衡数据集时能够提供相对鲁棒的分类结果(Japkowicz & Stephen，2002)。SVM 是通过以下方式促进学习的：通过使用邻近概念边界(支持向量)的具体实例，最大化支持向量与假设的概念边界(超平面)之间的分离间隔(软间隔最大化)，同时最小化总分类误差(Vapnik，1995)。

不平衡数据对 SVM 的影响利用了软间隔最大化模型的不足之处(Raskutti & Kowalczyk，2004；Akbani，Kwek & Japkowicz，2004)。由于 SVM 试图使总误差最小，这导致它们天生偏向于多数类概念。在最简单的情况下，一个两类空间可以由其多数类概念附近的“理想”分隔线线性分离。在这种情况下，可能会出现代表少数类概念的支持向量都“远离”这条“理想”的分隔线，因此，对最终分类器的贡献很少(Raskutti & Kowalczyk，2004；Akbani 等，2004；Wu & Chang，2003a)。而且，如果缺乏代表少数类概念的数据，将会出现代表性支持向量的不平衡，这也会降低分类性能。同样的特点在线性不可分空间里也是很明显的。在这种情况下，核函数将线性不可分空间映射到高维可分离空间。然而，这种情况下的最优超平面分类将偏向多数类，使得对多数类错误分类造成的高错误率降低到最小。最糟糕的情况下，SVM 将所有的样本分类为多数类——这是一种策略，如果数据分布严重不平衡的话，那么在整个数据空间里错误率将会达到最小。

2. 核方法与抽样法的集成

机器学习领域有很多文献把一般抽样与过/欠抽样 SVM 相集成(Viarino，Spyridonos，Radeva & Vitria，2005；Kang & Cho，2006；Liu，An & Huang，2006；Wang & Japkowicz，2008)。例如，SDC 算法对不同类使用不同的错误代价(Akbani 等，2004)来偏置 SVM，以便使决策边界远离阳性样本，从而使阳性样本更加密集地分布，以确保更加明确的边界。同时，由 Kang 和 Cho(2006) 以及 Liu 等(2006)提出的方法通过修改数据分布而不修改底层 SVM 分类器开发了一个集成系统。最后，Wang 和 Japkowicz(2008)提出以不对称错误分类代价修改 SVM 来增强其性能。这个想法类似于 AdaBoost. M1(Freund & Schapire，1996，2002)算法，后者使用一个迭代过程有效地修改训练观察值的权重，通过这种方式，可以基于连续学习过程来修改训练数据，从而提高分类性能。

细粒度支持向量机-重复欠抽样算法(GSVM-RU)是由 Tang 和 Zhang(2006)提出的，是 SVM 学习与欠抽样方法的集成。该算法所基于的细粒度支持向量机(GSVM)，是根据统计学习理论和粒度计算理论发展起来的(Tang，Jin & Zhang，2008；Tang，Jin，Zhang，Fang & Wang，2005；Tang，Zhang，Huang，Hum & Zhao，2005)。GSVM 的主要特征有两个：首先，GSVM 可以通过观察数据一个子集的局部意义与全局相关性之间的权衡，有效地分析数据的固有分布；其次，GSVM 通过使用并行计算提高了 SVM 的计算效率。在不平衡数据学习的情况下，GSVM-RU 方法通过使用一个将 SVM 本身用于欠抽样的迭代学习过程来充分利用 GSVM(Tang & Zhang，2006)。具体地，由于所有的少数(阳性的)样本被认为是富信息的，所以首先从这些样本中形成一个阳性信息粒，然后，利用阳性粒和数据集(即 S_{maj})中的剩余样本构建一个线性 SVM；阴性样本被这个 SVM 认为是支持向量，即所谓的“阴性局部支持向量”(NLSV)，这些向量形成阴性信息粒，并且从原始训练数据集中去除。接着构建一个新的线性 SVM，新一组的 NLSV 再次形成阴性粒，并从数据集中去除。此过程重复多次，以获得多个阴性信息粒。最后，考虑全局相关性，用一个聚合运算从那些迭代建立的阴性信息粒中选择特定的样本集，然后结合所有的阳性样本构建一个最终的 SVM 模型。以这种方式，GSVM-RU 方法利用 SVM 本身作为一种欠抽样机制，再用富信息的样本连续构建多个信息粒，最后再一起构建最终的 SVM。

3. 不平衡数据学习的核修正方法

除了上述的抽样法和基于核的学习方法的集成，另一种基于核方法的学习研究

工作更关注 SVM 本身的机制，这类方法通常被称为核修正方法。

核修正方法的一个例子是由 Hong、Chen 和 Harris(2008)提出的核分类器构造算法，该算法基于正交前向选择(OFS)和正则化正交加权最小二乘(ROWLS)估计量。该算法在基于核方法的学习模型中，通过引入处理两类数据集上不平衡数据分布的两个主要组件来优化算法的泛化性。第一个组件集成了留一法(leave-one-out，LOO)交叉验证概念和曲线下区域(AUC)评价指标(见 3.4.2 节)，以开发 LOO－AUC 目标函数，并将其作为最优化核模型的选择机制。第二个组件充分利用了 ROWLS 算法中的参数估计代价函数的代价敏感性，为少数类中的错误数据样本指派比多数类中的错误数据样本更大的权重。

核修正方法的其他实例是各种用于调节 SVM 类边界的技术。这些方法利用边界对齐技术提高支持向量机的分类性能(Wu & Chang，2003b，2004，2005)。例如，在 Wu 和 Chang(2003b)中，提出了 3 种调整边界偏斜的算法：边界运动(BM)方法、偏斜惩罚(BP)方法、类边界对齐(CBA)方法。此外，在 Wu 和 Chang(2004，2005)中，提出了一种核边界对齐(KBA) 算法，其基本思想是根据不平衡数据分布修改由核函数生成的核矩阵。KBA 方法的理论基础是自适应保角变换(ACT)，其中，核函数的保角变换是基于特征空间距离和类不平衡率的(Wu & Chang，2003b)。通过推广 ACT 的原理，KBA 方法通过修改特征空间的核矩阵，处理不平衡学习问题。理论分析和实验验证表明，该方法不仅提供了高准确性，也可以通过修改核矩阵将其应用到向量数据和序列数据中。

在更加集成化的基于核的学习方法中，Liu 和 Chen(2005，2007)提出了全间隔自适应模糊 SVM 核方法(TAF-SVM)，用于提高 SVM 的鲁棒性。TAF-SVM 的优点主要有 3 个。首先，TAF-SVM 通过“模糊化”训练数据处理过拟合问题，根据训练样本的相对重要性进行不同处理；其次，不同的代价算法被嵌入 TAF-SVM，使得这个算法能自适应不同的数据分布偏斜；最后，传统的软间隔最大化算法被全间隔算法取代，这在最优分类超平面的构造中同时考虑了错误分类和正确分类的数据样本。

关于不平衡数据学习的一个特别有趣的核修正方法是具有牛顿细化(Newton refinement)(Fung & Mangasarian，2005)的 k-类近似支持向量机(PSVM)。该方法基本上可以把软间隔最大化算法转化成适用于线性或非线性分类器的一个简单的 k 阶线性方程组，其中 k 是类的个数。这种方法的主要优点之一是执行学习过程非常快，因为它并不比解简单的线性方程组复杂。最后，在存在极度不平衡数据的情况

下，Raskutti 和 Kowalcyzk(2004)建议，如果 SVM 完全忽略其中的一个类，那么应该同时使用抽样和数据空间加权补偿技术。在这个过程中，为了平衡数据，使用两种平衡模式：一个相似度检测器用来学习基于阳性样本的鉴别，以及一个异常检测器用来学习基于阴性样本的鉴别。

机器学习领域还有一些其他的核修改学习方法，包括用于大规模不平衡数据集的支持聚类机(SCM)(Yuan，Li & Zhang，2006)、用于不平衡聚类的核 Neural-gas (KNG)算法(Qin & Suganthan，2004)、基于 k-邻近分类器和 P2P 通信方法的 P2PKNNC 算法(Yu & Yu，2007)，HKME 算法包括一个二元支持向量分类器(BSVC) 和一个具有高斯径向核函数的单类支持向量分类器(v-SVC)(Li，Chan & Fang，2006)，以及 Adaboost 相关向量机(RVM)(Tashk，Bayesteh & Faez，2007)等。此外，对于许多基于核的学习方法，在前面的"核方法与抽样法的集成"和"不平衡数据学习的核修正方法"小节中的两个主要类没有严格的区别。很多情况下的学习方法都利用核修改方法来改进性能。例如，Akbani 等(2004)以及 Wu 和 Chang (2003a)是不平衡学习混合解决方案的很好的实例。在本节中，为了更好地描述和组织内容，我们把基于核的学习方法分成了两个小节。

3.3.4 不平衡数据学习的主动学习方法

关于不平衡数据学习的主动学习方法也在机器学习领域得到了研究(He & Garica，2009)。传统上，主动学习方法主要用来解决无标记训练数据问题，然而，最近出现了一些关于不平衡数据集的各种主动学习问题的讨论(Abe，2003；Ertekin，Huang，Bottou & Giles，2007a；Ertekin，Huang & Giles，2007b；Provost，2000)。此外，应该指出的是，关于不平衡数据学习的主动学习方法往往被集成到基于核的学习方法中。因此，本小节和 3.3.3 节密切相关。

基于 SVM 的主动学习的目的是，从未知的训练数据中选择富信息样本来重新训练基于核的模型(Ertekin 等，2007b)，如那些最接近当前超平面的样本。图 3-9 说明了不平衡数据集的选择过程(Ertekin 等，2007a)。假设图 3-9 代表一个不平衡数据集的类分布，其中，阴影区域对应于间隔内的类分布。在这种情况下，间隔内的数据不平衡率比整个数据集的不平衡率小。受这种情况的启发，Ertekin 等(2007a，2007b)提出了一种有效的基于 SVM 的主动学习方法，用查询主动学习的每个迭代步长的一小部分数据代替查询整个数据集。在此过程中，在给定的训练数据集上训练 SVM，此后，根据建立的超平面提取富信息样本，并构建一个新的训练

集。最后，该过程使用这个新的训练集和所有未知的训练数据，并利用 LASVM 在线 SVM 学习算法，主动地再训练 SVM(Borders，Ertekin，Weston & Bottou，2002)，以促进主动学习过程。

Ertekin 等(2007a，2007b)还指出，富信息样本的搜索过程的计算代价很高，这是因为，对于未知的数据，该算法需要重新计算每个样本与当前超平面之间的距离。为了解决这个问题，他们提出了一个方法，以能够有效地从训练数据的一个随机集合中选择富信息样本，从而减少大规模不平衡数据集的计算成本(Ertekin 等，2007a，2007b)。此外，主动学习的提前停止准则也讨论了这项工作，相比随机样本选择解决方案，这项工作可以实现主动学习过程的快速收敛。

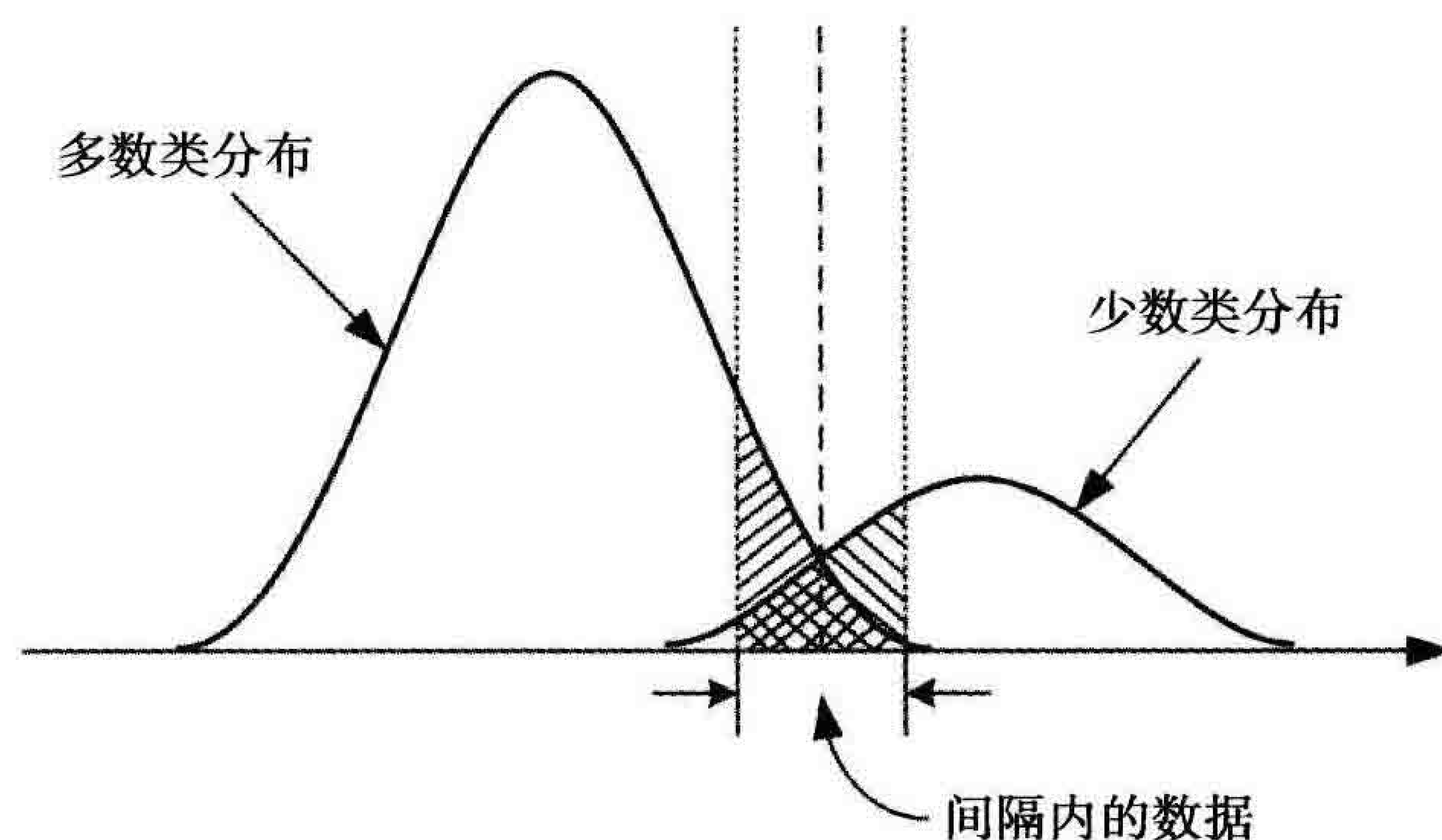


图 3-9 间隔内和间隔外的数据不平衡率(Ertekin 等(2007a，2007b))

除了基于核的集成，具有抽样技术的主动学习集成也得到了研究。例如，Zhu 和 Hovy(2007)分析了词义消歧(WSD)不平衡学习问题，在他们的工作中，主动学习方法的研究是基于不确定性抽样方法的，存在的挑战是如何测量未标记样本的不确定性，以便选择最大的不确定样本来增大训练数据。在这种情况下，熵作为不确定性的度量标准。此外，Zhu 和 Hovy(2007)中研究了基于最大置信度和最小误差的两个停止机制。仿真结果表明，可以用最大置信度和最小误差分别作为这种情况下主动学习停止条件的上界和下界。另一个主动学习的抽样方法是由 Doucette 和 Heywood(2008)提出的简单主动学习启发式(SALH)方法。这种方法的关键思想是，通过集成随机欠抽样方法和一个修改的 Wilcoxon-Mann-Whitney(WMW)代价函数(Doucette & Heywood，2008)，为遗传规划(GP)分类器的进化提供一个通用模型。SALH 方法的主要优点包括主动地为学习偏置数据分布、鲁棒代价函数的存在性以及改进与适应度评价相关的计算代价。在 6 个数据集上的仿真结果说明这个方法是有效的。

3.3.5 不平衡数据学习的其他方法

前面讨论了抽样方法、代价敏感方法、核方法和主动学习方法。值得注意的是，不平衡数据学习问题的解决方案并不局限于这几种方法，业界还从不同角度提出了一些其他的解决不平衡数据学习的方法。例如，对于不平衡学习，单类学习或异常检测方法也受到广泛关注(Chawla 等，2004)。一般来说，这类方法的目的在于通过使用主要样本或只使用一个单类样本(即基于识别的方法)来识别一个概念的实例，而不是像传统的学习方法(即基于区分的归纳方法)区分阳性和阴性类之间的实例。在这方面的代表性文献包括单类 SVM(Raskutti & Kowalczyk，2004；Scholkopf，Platt，Shawe-Taylor，Smola & Williamson，2001；Manevitz & Yousef，2001；Zhuang & Dai，2006a，2006b；Lee & Cho，2006)，以及自联想模型(或自动编码)方法(Japkowicz，2000，2001；Manevitz &Yousef，2007；Japkowicz，Myers & Gluck，1995)。特别地，Raskutti 和 Kowalcyzk(2004)指出单类学习方法在处理具有高维特征空间的极度不平衡数据集时非常有用。此外，Japkowicz(2001)提出了一个方法，可以训练一个自联想模型以在输出层重构阳性类，它表明在一定条件下，如在多模式域上，单类学习方法可能优于基于区分的方法。同时，Manevitz 和 Yousef(2001，2007)分别基于 SVM 和自动编码，给出了单类学习方法在文档分类领域中的成功应用。Japkowicz(2000)对不同抽样方法与单类自联想模型方法进行了比较，并给出了关于这两种方法的优点和克服其局限性的有用建议。Japkowicz 等人(1995)研究了基于冗余压缩和非冗余区分技术的异常检测。最近，Lee 和 Cho(2006)认为，异常检测对于极度不平衡数据集非常有用，然而，普通的基于区分的归纳分类器适用于相对温和的不平衡数据集。

最近，马田系统(Mahalanobis-Taguchi System，MTS)也被用于不平衡数据学习(Su & Hsiao，2007)。MTS 最初是一种多元数据诊断和预测技术(Taguchi，Chowdhury &Wu，2001；Taguchi & Jugulum，2002)。不像大多数分类算法，MTS 中的学习是使用单类样本而不是整个训练数据生成一个连续测量尺度。由于这种特性，MTS 模型可能不受偏态数据分布的影响，所以表现出鲁棒的分类性能。Su 和 Hsiao(2007)对不平衡数据学习的 MTS 模型进行了评估，并与逐步判别分析(SDA)、反向传播神经网络、决策树以及 SVM 进行了比较。结果表明，在处理不平衡数据时，MTS 方法是有效的。而且，Su 和 Hsiao(2007)提出了一种基于切比雪夫定理的概率阈值方法，能够用来确定合适的 MTS 分类阈值。

另一个重要问题是3.2节中所讨论的不平衡数据与小样本问题的结合，针对这一问题，Caruana(2000)提出了两种解决方法。第一种方法建议用秩度量(rank metric)代替传统的准确性度量来作为训练和模型选择的标准。基于秩度量的方法更加注重学习区分类本身而不是类的内部结构(特征空间连接)，这有利于促进对于具有小样本和高维度的不平衡数据集的学习。第二种方法是基于多任务学习的方法，其基本思想是使用一个共享的数据表示来训练与主要任务相关的额外任务模型，从而添加额外的训练信息到数据中，扩大所表示类的有效大小(Caruana，2000)。

最后，还要注意的是，虽然目前的工作都集中在两类不平衡数据学习问题上，但是多类不平衡数据学习问题是同等重要的。例如，Sun等(2006)提出一个代价敏感自举算法——AdaC2.M1，用以解决多类不平衡数据学习问题。在他们的研究中，采用遗传算法搜索每个类的最优代价设置。Abe等(2004)提出一种多类代价敏感学习的迭代方法，该方法基于3个主要观点：迭代代价加权、数据空间扩展和随机梯度自举。Chen等(2006)提出一种最小-最大模块化网络，将多类不平衡数据学习问题分解为一系列小的两类分类子问题。其他多类不平衡数据学习工作包括多类代价敏感神经网络的重标度方法(Zhou & Liu，2006；Liu & Zhou，2006b)、不平衡样本集(eKISS)的集成方法(Tan等，2003)等。

很明显，现存的不平衡数据学习问题的解决方案是多层面而且相互关联的，因此，这些解决方案的评价技术具有相似的特征。下面的章节将重点讨论不平衡数据学习的评价指标。

3.4 不平衡数据学习的评价指标

由于在机器学习领域出现了越来越多复杂且有潜力的不平衡数据学习算法，所以制定标准化的评价指标来评价这些算法的有效性是非常必要的。本节主要讨论不平衡数据学习的主要评价指标(He&Carcia，2009)。

3.4.1 单一评价指标

传统上，最常用的评价指标是准确度和误差率。考虑一个基本的两类分类问题，设$\{p, n\}$为真阳性类和真阴性类标签，$\{Y, N\}$为预测的阳性类和阴性类标签，分类性能可以表示为混淆矩阵(列联表)，如图3-10所示。

本章用少数类作为阳性类，多数类作为阴性类，根据这个约定，准确度和误差率的定义如下：

分类器输出	真实类 p	真实类 n
Y	TP（真阳性）	FP（假阳性）
N	FN（假阴性）	TN（真阴性）
列计数:	P_c	N_c

图 3-10 性能评估的混淆矩阵

$$\text{准确度} = \frac{\mathrm{TP}+\mathrm{TN}}{P_C+N_C} \tag{3-13}$$

$$\text{误差率} = 1-\mathrm{accuracy} \tag{3-14}$$

这些指标提供了一种描述给定数据集上分类器性能的简单方法。然而，在某些情况下，这些指标可能不可靠并且对数据的变化高度敏感。在最简单的情况下，如果一个给定的数据集包括5%的少数类样本和95%的多数类样本，简单地说，将每个样本分类为多数类样本的准确度应该为95%。从表面上看，在整个数据集上，95%的准确度已经表现出了很好的性能，然而，这个描述未能反映所识别出的少数类样本为0%这个事实。也就是说，在这种情况下，准确度指标没有提供关于分类器分类类型的充分信息。

关于不平衡数据学习，现存的一些研究文献中包含了无效的准确度（He & Garcia，2008；Guo & Viktor，2004b；Weiss，2004；Chawla，2003；Maloof，2003；Sun 等，2007；Joshi，Kumar & Agarwal，2001；Provost & Fawcett，1997；Provost，Fawcett & Kohavi，1998）。图3-10中的混淆矩阵可以解释无效准确度的根本问题：左列表示数据集的阳性样本，右列表示阴性样本，因此，两列数据的比例代表数据集的类分布，使用这两列值的任何指标将对数据分布不平衡固有敏感。从等式(3-13)可以看出，准确度使用了两列信息。因此，当类分布变化时，性能的测量将会改变，即使分类器的基本性能没有改变。可以想象，在不同的数据集上比较不同学习算法的性能时，由于性能表现的不一致性，所以会存在很大的问题。换句话说，在不平衡数据存在的情况下，当评价指标对数据分布敏感时，算法性能的对比分析将变得难以进行。

该领域经常采用其他评价指标代替准确度以对不平衡数据学习问题进行综合评价，即查准率(precision)、查全率(召回率，recall)、F-度量(F-measure)和G-均值(G-mean)。这些指标的定义如下：

$$查准率 = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3-15}$$

$$查全率 = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-16}$$

$$F\text{-}度量 = \frac{(1+\beta)^2 \cdot 查全率 \cdot 查准率}{\beta^2 \cdot 查全率 + 查准率} \tag{3-17}$$

其中，β是一个系数，用于调整查准率对查全率的相对重要性(通常，$\beta=1$)。

$$G-均值 = \sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}} \tag{3-18}$$

直观地看，查准率是精确性的度量(即，在标记为阳性的样本中，实际上有多少个样本被正确地标记)，而查全率是完备性的度量(即，有多少阳性类样本被正确标记)。这两个指标很像准确度和误差率，共享彼此之间的互逆关系。然而，又不像准确度和误差率，查准率和查全率对数据分布的变化并不都是敏感的。从查准率和查全率的公式容易看出，查准率(见式(3-15))对数据分布敏感，而查全率(见式(3-16))对数据分布不敏感。另一方面，所谓的查全率不依赖于数据分布，几乎是多余的，这是因为查全率无法反映出有多少样本被错误地标记为阳性，从而导致了仅仅根据查全率做出的判断是有歧义的。同样，查准率不能判断有多少阳性样本被错误地标记。然而，如果使用得当，查准率和查全率可以有效地评价不平衡数据学习情况下的分类性能。特别地，查准率和查全率可以组合产生一个单一的值，称为F-度量指标(见式(3-17))，它根据用户设置的系数β确定查全率或查准率的比例，作为分类有效性的度量。与准确度指标相比，F-度量对分类器的功能提供了更深层的描述，而且还保留了对数据分布的敏感性。另一个指标，G-均值指标(见式(3-18))根据阳性准确度与阴性准确度的比率，评价归纳偏斜的程度。虽然，F-度量和G-均值在很大程度上提高了准确度，但是在解决关于分类评价的更一般性的问题方面依然是无效的。例如，如何才能在一个宽泛的样本分布范围内比较不同分类器的性能？

3.4.2 受试者工作特性(ROC)曲线

为了解决这些问题，ROC评价技术(Fawcett，2003，2006)采用两个基于单列的评价指标的比例，即真阳性率(TP_rate)和假阳性率(FP_rate)，定义为：

$$\mathrm{TP_rate} = \frac{\mathrm{TP}}{P\mathrm{c}}; \quad \mathrm{FP_rate} = \frac{\mathrm{FP}}{N\mathrm{c}} \tag{3-19}$$

ROC图是描述假阳性率与真阳性率之间关系的曲线，ROC空间上的任何点对应于单一分类器关于给定数据分布的性能。ROC曲线提供了分类的收益(通过真阳

性反映）与代价（通过假阳性反映）之间相互权衡的可视化表示。

对于仅仅输出离散类标签的硬型分类器，每个分类器产生对应于 ROC 空间的单个点的真阳性率和假阳性率。图 3-11 举例说明了一个典型的 ROC 图，点 A、B、C、D、E、F 和 G 代表 ROC 点，曲线 L_1 和 L_2 代表 ROC 曲线（He & Garcia，2009）。根据 ROC 图的结构，点 $A(0，1)$代表一个完美的分类。一般来说，如果一个分类器在 ROC 空间中的对应点比另一个分类器的对应点更接近点 A（在 ROC 空间左上角），说明这个分类器比另一个分类器的性能更好。如果一个分类器对应的 ROC 点位于对角线上，如图 3-11 中的点 E，则表示该分类器对类标签是随机猜测的（即，一个随机分类器）。因此，如果一个分类器的 ROC 曲线位于 ROC 空间右下角的直角三角形区域内，则表示其性能比随机猜测更加糟糕，如图 3-11 中的阴影区域内与 F 点相关联的分类器。然而，这并不意味着一个比随机猜测更糟糕的分类器不能提供有用的信息。相反，这个分类器是富信息的，但是，信息未能被正确利用。例如，如果否定分类器 F 的分类结果，即倒转对每个样本的分类决策，那么将产生与图 3-11 中的 F 点对称的分类点 G。

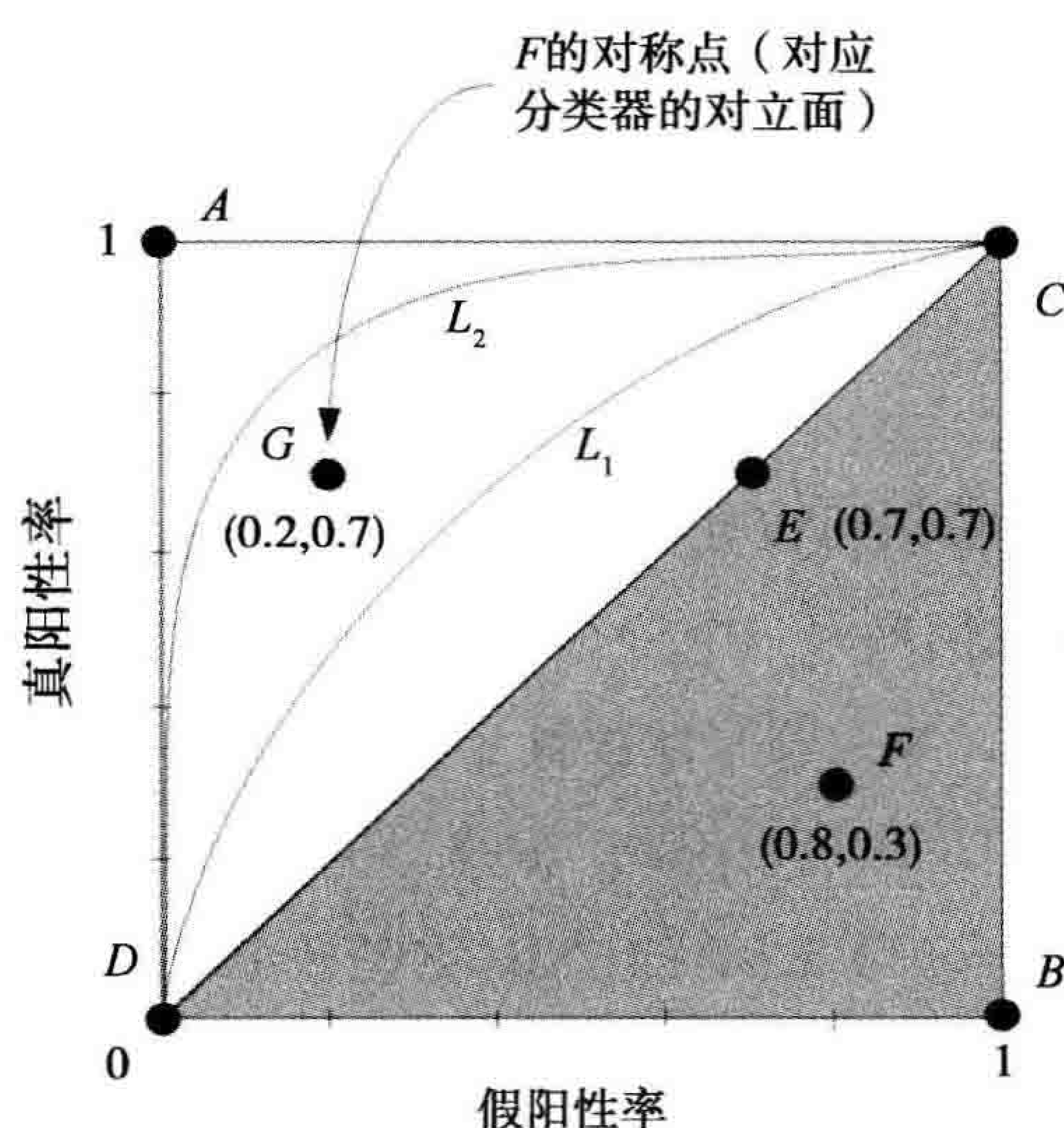


图 3-11　ROC 曲线

在软型分类器的情况下，即分类器输出一个连续数值来代表一个样本属于预测类的置信度，可以用一个阈值来生成 ROC 空间上的点。

这种技术可以产生 ROC 曲线，而不是单一的 ROC 点，如图 3-11 中的曲线 L_1 和 L_2。为了在这种情况下评价不同分类器的性能，通常使用曲线下的面积（AUC）作为评价指标（Fawcett，2003，2006）。例如，在图 3-11 中，L_2 曲线比 L_1 曲线的

AUC大，因此，L_2 曲线对应的分类器可以提供比 L_1 曲线对应的分类器更好的平均性能。当然，应该注意，在ROC空间的特定区域内，高AUC分类器可能比低AUC分类器的性能更糟糕(Fawcett，2003，2006)。另外还请注意，基于分类器的内在特性，硬型分类器通常提供软型输出。(Domingos，1999；Freund & Schapire，1996；Provost& Domingos，2000；Fawcett，2001)。

3.4.3 查准率-查全率(PR)曲线

尽管ROC曲线提供了强有力的方法来可视化性能评估，但是它也存在自身的局限性。在高偏斜数据集的情况下，可以看出ROC曲线可能会对算法性能提供一个过于乐观的评价。在这种情况下，PR曲线可以对算法的性能评价提供富信息表示(Davis & Goadrich，2006)。

考虑图3-10中的混淆矩阵以及查准率和查全率的定义，PR曲线反映了查准率与查全率之间的关系。PR曲线与ROC曲线具有很强的一致性：一条曲线在ROC空间优越，当且仅当它在PR空间优越(Davis & Goadrich，2006)。然而，优化ROC空间的AUC的算法并不能保证优化PR空间的AUC(Davis & Goadrich，2006)。此外，虽然ROC曲线的目标位于ROC空间的左上角，但是优势PR曲线位于PR空间的右上角。PR空间也具有类似于ROC空间的凸壳，即有意义的PR曲线(Davis & Goadrich，2006)。因此，PR空间与ROC空间具有类似的性能，也是一种有效的评价方法。

为了说明为什么PR曲线可以在高度不平衡数据的情况下，为算法的性能评价提供丰富的信息表示，我们考察一个分布，其中阴性样本显著超过阳性样本(即 $Nc >> Pc$)。在这种情况下，如果假阳性数量变化很大，由于分母(Nc)很大，FP率将不会有明显的改变(见式(3-19))，因此，ROC图将无法捕捉这一现象。另一方面，考虑查准率指标，即TP与TP+FP的比率(见图3-10和式(3-15))，当假阳性的数量大幅度改变时，它可以正确捕捉分类器的性能(Davis & Goadrich，2006)。这个例子明确地说明了在存在高偏斜数据时，PR曲线对于算法的性能评价是一种优势技术。因此，在该领域，很多当前的研究工作都在使用PR曲线进行算法的性能评价和比较(Bunescu等，2005；Davis等，2005；Singla & Domingos，2005；Landgrebe，Paclik，Duin & Bradley，2006)。

3.4.4 代价曲线

ROC曲线的另一个缺点是不能提供关于分类器性能的置信区间，无法推断不同

分类器性能的统计显著性(Holte & Drummond, 2005, 2006),而且难以根据类别概率或错误分类代价提供分类器性能的评价描述(Holte & Drummond, 2005, 2006)。为了提供一个更全面的评价指标来解决这些问题,Holte 和 Drummond (2000, 2005, 2006)提出了代价曲线的概念。代价曲线是一种代价敏感评价技术,它能够以可视化的方式,从错误分类代价和类分布方面,表达一个分类器的性能。因此,代价曲线方法保留了 ROC 分析具有吸引力的可视化表现形式,并提供了关于分类性能的更广泛的信息。

一般来说,代价曲线方法在工作点上表现算法的性能(即,归一化预期代价),这由概率代价函数(probability cost function)表示。概率代价函数表示正确分类一个阳性样本的概率。代价空间表现了 ROC 空间的二元性,ROC 空间中的点表示为代价空间中的线,反之亦然(Holte & Drummond, 2006)。在 ROC 空间中的任何(FP, TP)分类对对应于代价空间中的一条线:

$$E(C) = (1 - \mathrm{TP} - \mathrm{FP})\mathrm{PCF}(+) + \mathrm{FP} \tag{3-20}$$

其中,$E(C)$是预期代价,PCF(+)是一个样本来自阳性类的概率。图 3-12 给出了一个代价空间的例子。在图 3-12 中,我们强调了图 3-11 中的 ROC 点与其在代价空间中线的对应关系。例如,底部轴表示完美分类,而顶部轴表示相反的情况,它们分别对应于 ROC 点 A 和 B。

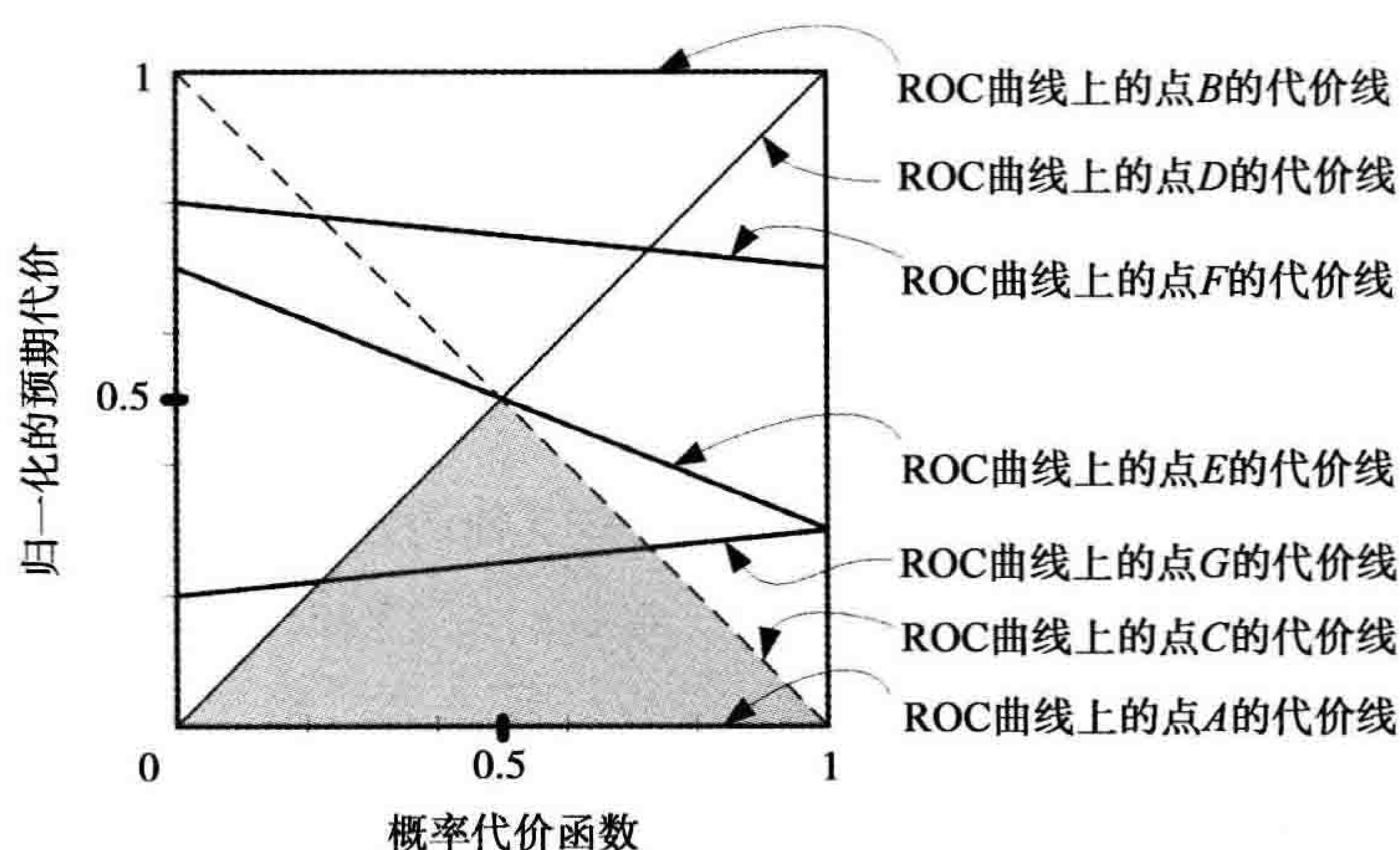


图 3-12 代价曲线

利用这些代价线,可以通过为每个可能的工作点选择一条分类线,从而建立代价曲线。例如,通过对所有可能的工作点最小化其归一化的预期代价,可以创建代价曲线。特别地,与 ROC 曲线相比,代价曲线技术对于分类性能具有更清楚的可视化表示,并且可以在分类器之间进行更直接的评价。

3.4.5 多类不平衡数据学习评价指标

虽然到目前为止，本节所讨论的评价指标都只适合于两类不平衡数据学习问题，但是，其中的一些可以被修改为适应多类问题，例如，Fawcett(2003，2006)中讨论的多类 ROC 图。对于一个 n 类问题，图 3-10 所示的混淆矩阵可以扩展为一个 $n\times n$ 矩阵，其中有 n 个正确分类(主对角线上的元素)和 n^2-n 个错误分类(非主对角线上的元素)。因此，不能只表示单一收益(TP)与代价(FP)之间的权衡，而应该描述 n 个收益与 n^2-n 个代价之间的关系。要想实现这样的描述，一个简单方法是对每个类生成 n 个不同的 ROC 曲线图(Fawcett，2003，2006)。例如，考虑一个共有 W 个类的分类问题，类 ω_i 作为阳性类，所有其他类作为阴性类，用第 i 个 ROC 图 ROC_i 表示分类性能。那么，这种方法减弱了用 ROC 分析不平衡数据学习问题的主要优势：由于这种情况下的阴性类是 $n-1$ 个类的组合，所以这种方法对类偏斜很敏感(参照 3.4.1 和 3.4.2 节)。

同样，在多类不平衡数据学习场景中，两类问题中的 AUC 值可以推广为多对值(Hand & Till，2001)。为了计算这样的多类 AUC，Provost 和 Domingos 提出了一种基于概率估计的方法。首先，生成每个参考类 ω_i 的 ROC 曲线，测量它们各自的 AUC。然后，根据参考类在数据中的出现率确定权重系数，通过权重系数组合所有的 AUC。虽然这种方法在计算上相当简单，但是如前所述，它对类偏斜是敏感的。为了消除这一缺陷，Hand 和 Till(2001)提出了 M-度量，M-度量是一种广义方法，该方法基于 AUC 的内在特性，把所有类对聚集在一起。这种方法的主要优点是对类分布和错误代价不敏感。有兴趣的读者可以参阅文献(Hand & Till，2001)了解关于这种技术的详细细节。

除了多类 ROC 分析，学术界还采用了一些其他的评价指标来分析多类不平衡数据学习问题。例如，在代价敏感学习中，很自然地把错误分类代价应用于多类不平衡数据学习问题的性能评估(Abe 等，2004；Zhou & Liu，2006；Liu & Zhou，2006b)。同时，Sun 等(2006)将 G-均值的定义(见式(3-18))推广到多类不平衡数据学习问题，用于计算其中的每个类的召回值的几何平均。

3.5 机遇和挑战

当今，随着原始数据的可用性不断以爆发式的速度增长，不平衡数据学习在很多领域起着至关重要的作用，并给相应领域的研究带来了新的机遇与挑战。然而，

在新的挑战出现的同时，还需要人们关注于对基础理论的理解，并建立基本的方法来解决一些挑战性问题。在本节中，我们简要概述不平衡数据学习的重要机遇和挑战，更加详细的讨论参见文献(He 和 Garcia，2009)。

第一，关于不平衡数据学习，当前大多数研究工作都集中在特殊算法和/或个案研究上，只有非常有限的研究工作阐述了对这个问题的基本原理和结论的理论性解释(He & Garcia，2009)(Provost，2000)。例如，虽然几乎现存文献中的每个算法都声称可以把分类准确度提高到某个水平，但是有些情况下，从原始数据集中学习可以提供更好的性能。这就产生了一个重要问题：不平衡数据学习方法对学习能力有多大的帮助？这是该领域的一个基本而且重要的问题，原因如下(He & Garcia，2009)：首先，假设存在特定的(现有或未来)技术或方法，它们在大多数(或者，在理想情况下，所有的)应用领域明显优于其他的技术或方法，那么对这些方法的深入研究将有益于对当前问题的基本理解。其次，随着数据工程研究方法逐步成为真实环境中实际问题的解决方案，一些问题，如“这个解决方案将提供怎样的帮助？”或“这个解决方案可以有效地处理各种类型的数据吗?”，成为做出经济和行政决策的基础。因此，这些关键问题已经广泛影响了机器学习领域的发展和数据工程的发展(He & Garcia，2009)。这些问题与 Provost 在 AAAI 2000 研讨会上关于不平衡数据集的特邀报告(Provost，2000)所提出的重要命题密切相关，“[关于不平衡数据学习,]…专注于机器学习算法如何能够最有效地处理任何给定数据？这并不是最好的策略。”我们相信，对于不平衡数据学习的基础理论的理解，不仅能为解决不平衡数据学习问题提供基本思路，而且还能提供比现有研究结果更具优势的技术方法。

第二，由于数据资源对于知识发现和数据工程领域的研究发展至关重要，关于不平衡数据学习问题的严格、有效的基准将对该领域的长期研究和发展非常有益。目前，尽管存在一些基准可以对数据工程算法/工具的有效性进行评价，但是这些基准的数量非常有限，即便存在，也很少有专门针对不平衡数据学习问题的基准或标准。例如，许多现存的评价基准不能明确地识别不平衡数据集，因此，在应用到不平衡数据学习之前，需要对数据集进行一些额外的处理。这个限制可能会阻碍不平衡数据学习的长足发展，因为缺乏性能评价的统一基准，以及缺乏数据共享和数据互操作性。

第三，建立标准化的评估实践，包括更多的评价指标，如3.4节所描述的基于曲线的评价技术，是非常必要的。正如3.4.1节所讨论的，传统技术使用单一的评价指标，如总准确度或总误差率，这对处理不平衡数据学习问题是不充分的。虽然

大多数文献通过组合多单一评价指标来评价算法的性能及其权衡，但是，如果没有伴随基于曲线的分析，如在 3.4.2 节、3.4.3 节和 3.4.4 节中所讨论的那样，那么很难对不同算法进行具体的比较评价，也很难回答关于算法功能方面的更严谨的问题。因此，标准化的评估实践将有利于该领域的长期研究发展。这不仅是因为每种技术对于不同的基本问题有自己的一套应对方案，也是因为一种技术的评价空间中的分析可以关联到其他评价空间中，从而增加了评价的传递性，以及对现有和未来研究工作的更宽泛理解。

第四，研究不平衡数据的新兴应用是很重要的。例如，如何处理具有不平衡数据的增量学习问题？在这种情况下，如果在学习过程的中期引入不平衡数据，那么机器学习系统如何自主调整学习算法？如果新引进的概念也是不平衡的，那么如何处理这种情况(即不平衡概念漂移问题)？另一个有趣的例子是来自不平衡数据的半监督学习。简单地说，半监督学习主要考虑数据集是标记和未标记数据的组合时的学习问题，相反，全监督学习考虑的是所有训练数据都是标记数据。在半监督学习的情况下，如何识别一个未标记的数据样本是来自平衡的还是不平衡的分布？什么是利用有标记的不平衡训练数据恢复未标记数据样本的有效且高效的方法？我们相信，所有这些问题不仅对理论研究的发展而且对许多实际应用都是重要的。

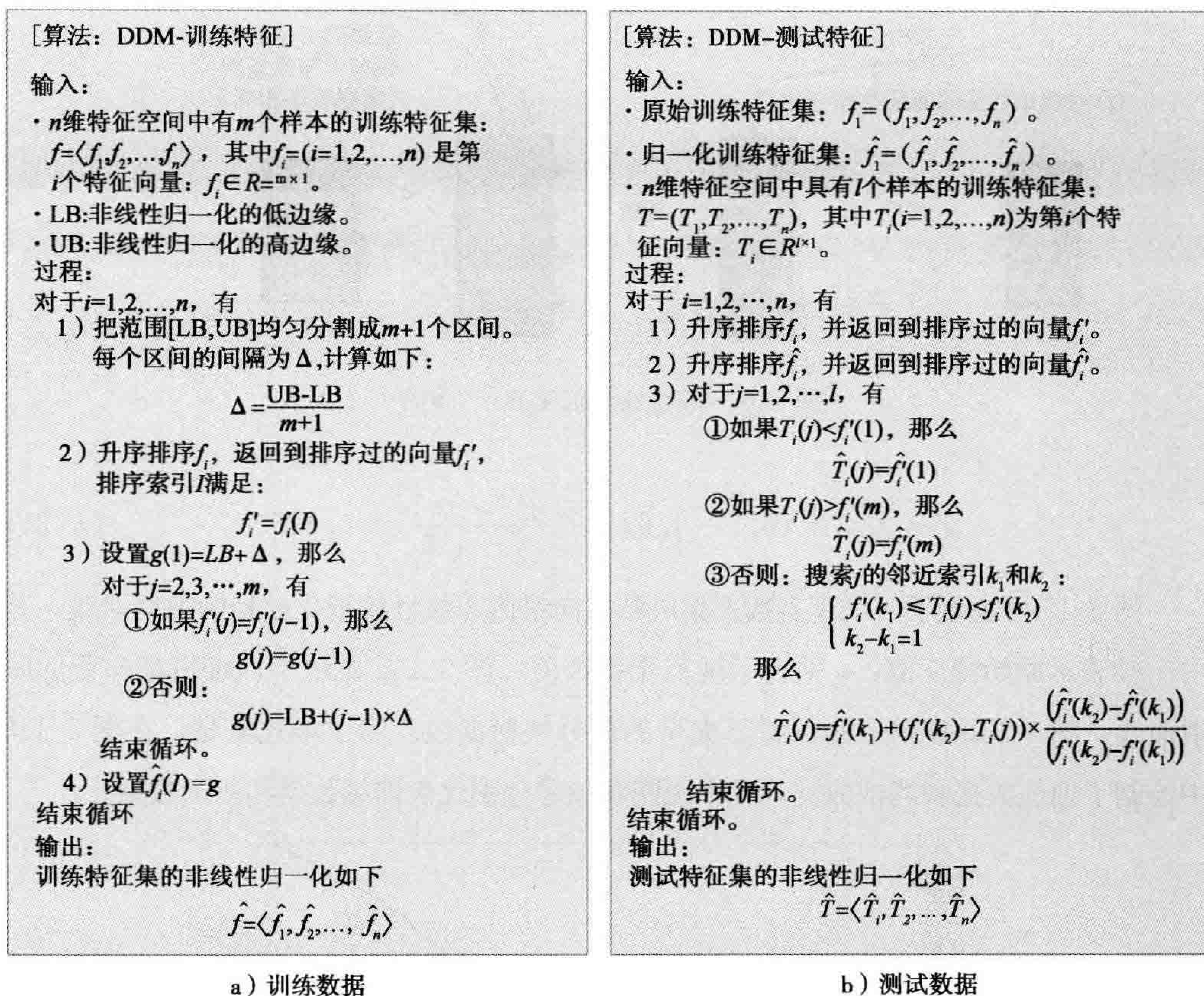
3.6 实例研究

本节基于多个基准，描述了一个不平衡数据学习的研究实例。特别地，我们讨论了 7 个学习方法的仿真实验，包括 RAMOBoost、SMOTEBoost、SMOTE、ADASYN、AdaCost、BorderlineSMOTE 和 SMOTE-tomek。

3.6.1 非线性规范化

数据规范化是许多学习算法的重要预处理步骤。例如，不同采集时间段数据(如训练数据和测试数据)的误匹配，可能会严重降低分类的性能。Viikki、Bye 和 Laurila(1998)指出，不同周期的噪声将会显著影响语音识别系统的性能。此外，规范化方法假设训练数据和测试数据在整个学习阶段都是有效的，并且在规范化之前，把训练特征集和测试特征集联合在一起搜索最大值和最小值。然而，在很多实际应用场景中，通常的现象是，在训练期间，可能拿不到测试数据。本节提出了一种基于数据分布的非线性规范化方法、分布距离映射(DDM)方法。在训练阶段，对于每一个特征，首先排序所有的训练数据，并且根据其在排序数组中的位置给它分配一

个非递减值，该值来自统一划分的区间[LB，UB]。这里，LB和UB代表用户定义的下界(LB)和上界(UB)，用于指定规范化过程后的数据范围，在当前的仿真实验中定义为[0，1]。以这种方式，建立一个单调序列，作为训练数据缩放比例。在测试阶段，首先寻找每个测试样本所落在的相应训练数据的区域，然后使用局部线性拟合函数来获取每个测试数据的规范化值。DDM方法的完整伪码如图3-13所示。



[算法：DDM-训练特征]

输入：

- n维特征空间中有m个样本的训练特征集：$f=\langle f_1,f_2,\ldots,f_n\rangle$，其中$f_i=(i=1,2,\ldots,n)$是第$i$个特征向量：$f_i\in R^{m\times 1}$。
- LB:非线性归一化的低边缘。
- UB:非线性归一化的高边缘。

过程：

对于$i=1,2,\ldots,n$，有

1）把范围[LB,UB]均匀分割成$m+1$个区间。每个区间的间隔为Δ,计算如下：

$$\Delta=\frac{\text{UB-LB}}{m+1}$$

2）升序排序f_i，返回到排序过的向量f_i'，排序索引I满足：

$$f_i'=f_i(I)$$

3）设置$g(1)=LB+\Delta$，那么

对于$j=2,3,\cdots,m$，有

①如果$f_i'(j)=f_i'(j-1)$，那么

$$g(j)=g(j-1)$$

②否则：

$$g(j)=\text{LB}+(j-1)\times\Delta$$

结束循环。

4）设置$\hat{f}_i(I)=g$

结束循环

输出：

训练特征集的非线性归一化如下

$$\hat{f}=\langle\hat{f}_1,\hat{f}_2,\ldots,\hat{f}_n\rangle$$

[算法：DDM-测试特征]

输入：

- 原始训练特征集：$f_1=(f_1,f_2,\ldots,f_n)$。
- 归一化训练特征集：$\hat{f}_1=(\hat{f}_1,\hat{f}_2,\ldots,\hat{f}_n)$。
- n维特征空间中具有l个样本的训练特征集：$T=(T_1,T_2,\ldots,T_n)$，其中$T_i(i=1,2,\ldots,n)$为第i个特征向量：$T_i\in R^{l\times 1}$。

过程：

对于$i=1,2,\cdots,n$，有

1）升序排序f_i，并返回到排序过的向量f_i'。

2）升序排序$\hat{f}_i$，并返回到排序过的向量$\hat{f}_i'$。

3）对于$j=1,2,\cdots,l$，有

①如果$T_i(j)<f_i'(1)$，那么

$$\hat{T}_i(j)=\hat{f}_i'(1)$$

②如果$T_i(j)>f_i'(m)$，那么

$$\hat{T}_i(j)=\hat{f}_i'(m)$$

③否则：搜索j的邻近索引k_1和k_2：

$$\begin{cases}f_i'(k_1)\leqslant T_i(j)<f_i'(k_2)\\k_2-k_1=1\end{cases}$$

那么

$$\hat{T}_i(j)=\hat{f}_i'(k_1)+(f_i'(k_2)-T_i(j))\times\frac{(\hat{f}_i'(k_2)-\hat{f}_i'(k_1))}{(f_i'(k_2)-f_i'(k_1))}$$

结束循环。

结束循环。

输出：

测试特征集的非线性归一化如下

$$\hat{T}=\langle\hat{T}_1,\hat{T}_2,\ldots,\hat{T}_n\rangle$$

a）训练数据　　b）测试数据

图3-13　非线性规范化方法：DDM算法

例如：为了更好地说明DDM非线性规范化方法的思想，图3-14给出了一个简单的例子来说明其操作过程。对于多维数据，DDM方法需要根据图3-13中的伪码，相应地规范化每个特征。为了清楚地说明，我们考虑6个数据样本，并用单一特征作为例子。假设规范化数据范围为[0，1](LB＝0，UB＝1)。在训练阶段，特征值1.3是排序过的训练数据向量中的第一个数据，因此，它的规范化值等于0.1667。完成训练数据规范化之后，就可以根据每个测试数据在训练数据的排序序列中的位置计算其规范化值。例如，测试数据6.7位于训练数据4.6和7.4之间，通过使用图3-13b中的内插公式，6.7的规范化值为

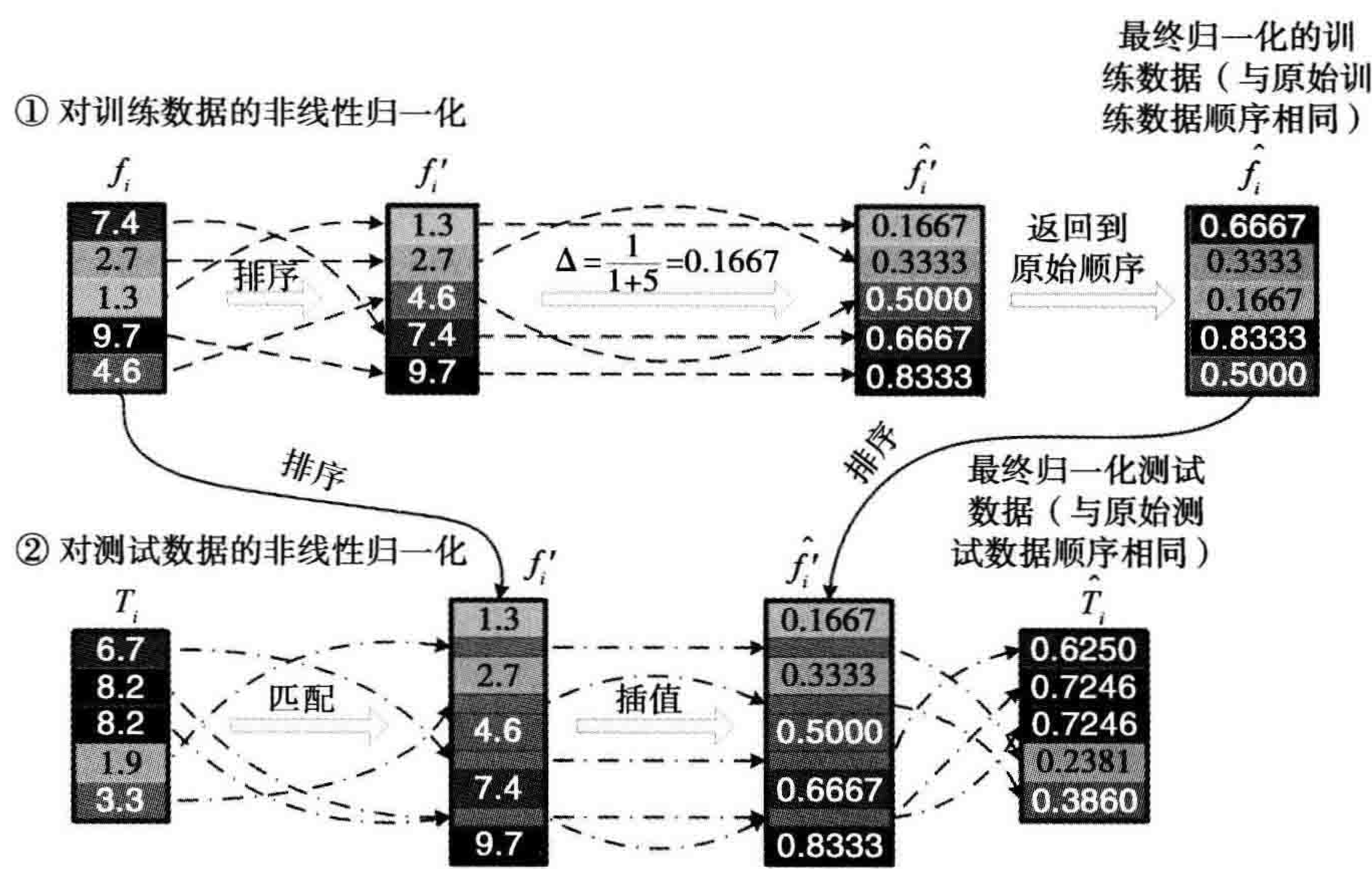
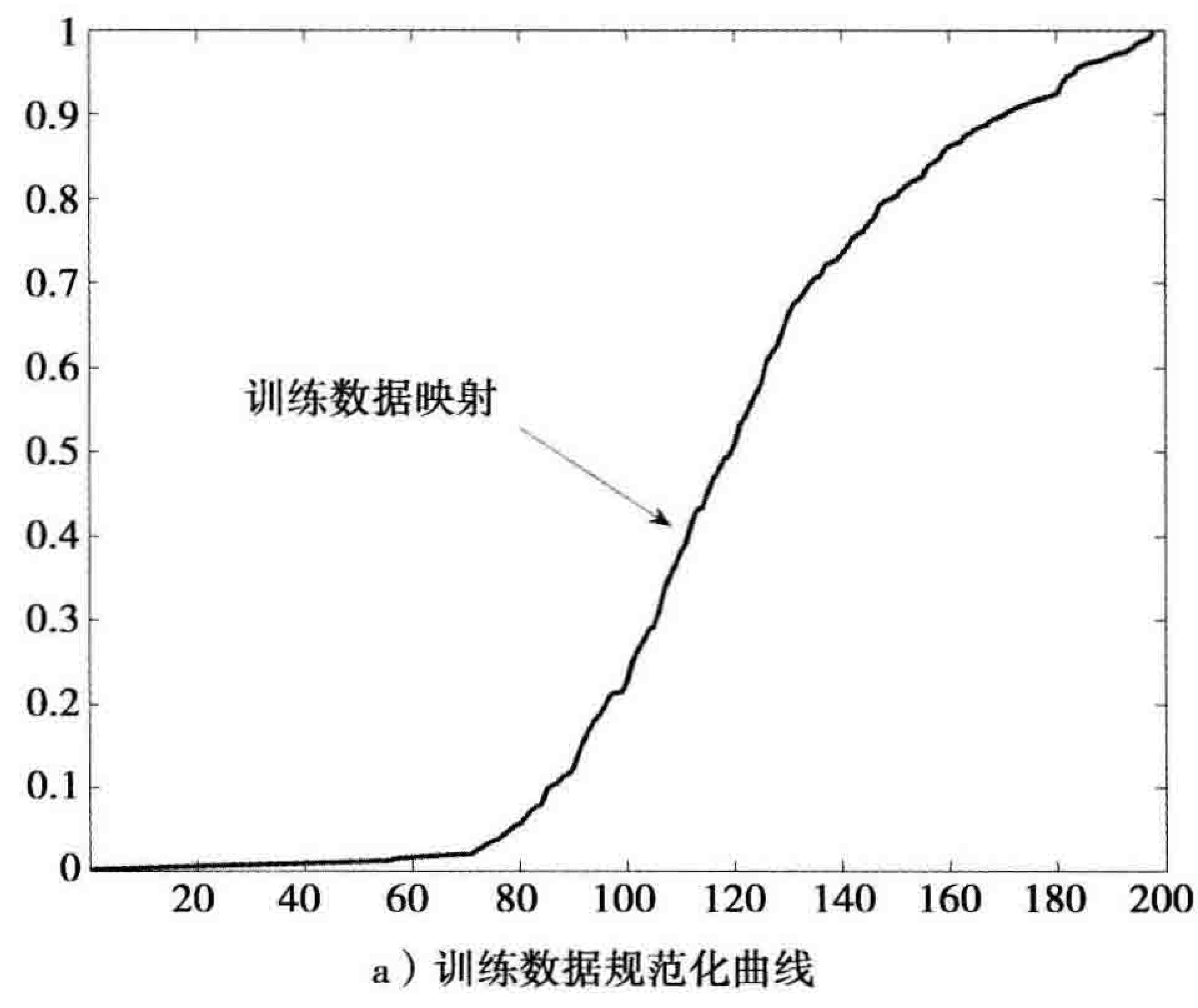


图 3-14 非线性规范化的一个例子

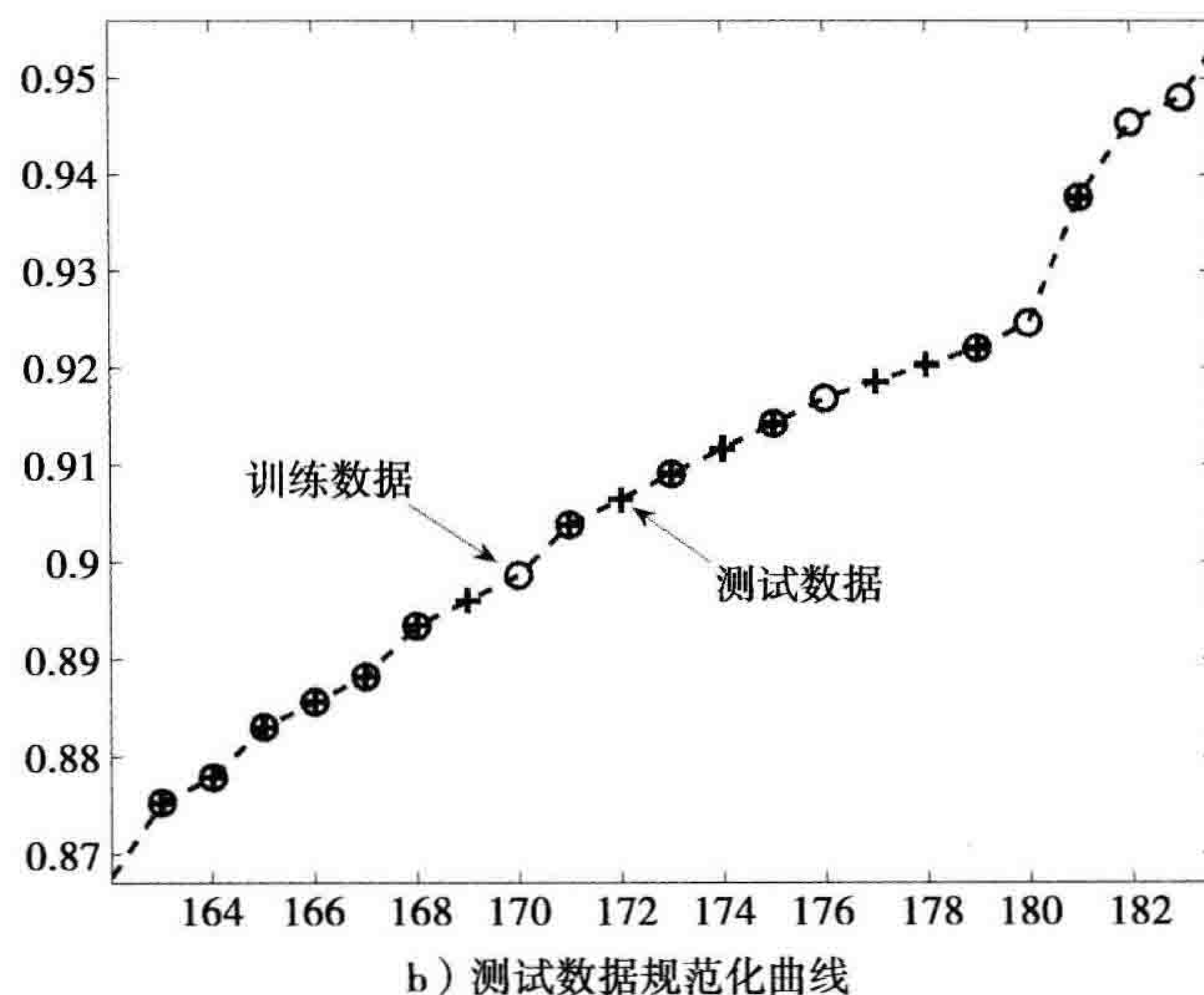
$$T=0.5+(6.7-4.6)\times\frac{0.667-0.5}{7.4-4.6}=0.6250 \tag{3-21}$$

图 3-15 可视化了一个真实数据集的特征向量的非线性规范化过程的映射曲线，其中 x 轴表示原始特征值，y 轴表示规范化特征值。图 3-15a 显示了训练特征向量的映射曲线，图 3-15b 显示了测试特征向量的部分映射曲线。为了描述清楚，在图 3-15b 中绘制了训练数据和测试数据，其中圆圈和加号分别代表训练数据和测试数据。



a）训练数据规范化曲线

图 3-15 “皮马人-印第安人-糖尿病人”(PID)数据集的第二特征向量的非线性规范化过程的映射曲线



b）测试数据规范化曲线

图 3-15 （续）

类似于在图像处理中利用直方图均衡化的概念提高图像的对比度并增强图像细节(Gonzalez & Woods，2002)，非线性规范化方法也可以平衡数据集的偏态分布，从而促进学习。与此相反，传统的线性规范化方法并没有这样的能力。为了观察使用规范化方法后的数据分布特性，图 3-16a 显示了一个测试特征向量规范化之前的直方图，图 3-16b 和 c 分别代表了使用传统线性规范化方法和 DDM 方法之后的直方图。从图 3-16a 可以看到原始测试特征向量的直方图主要位于[80，200]范围内，这

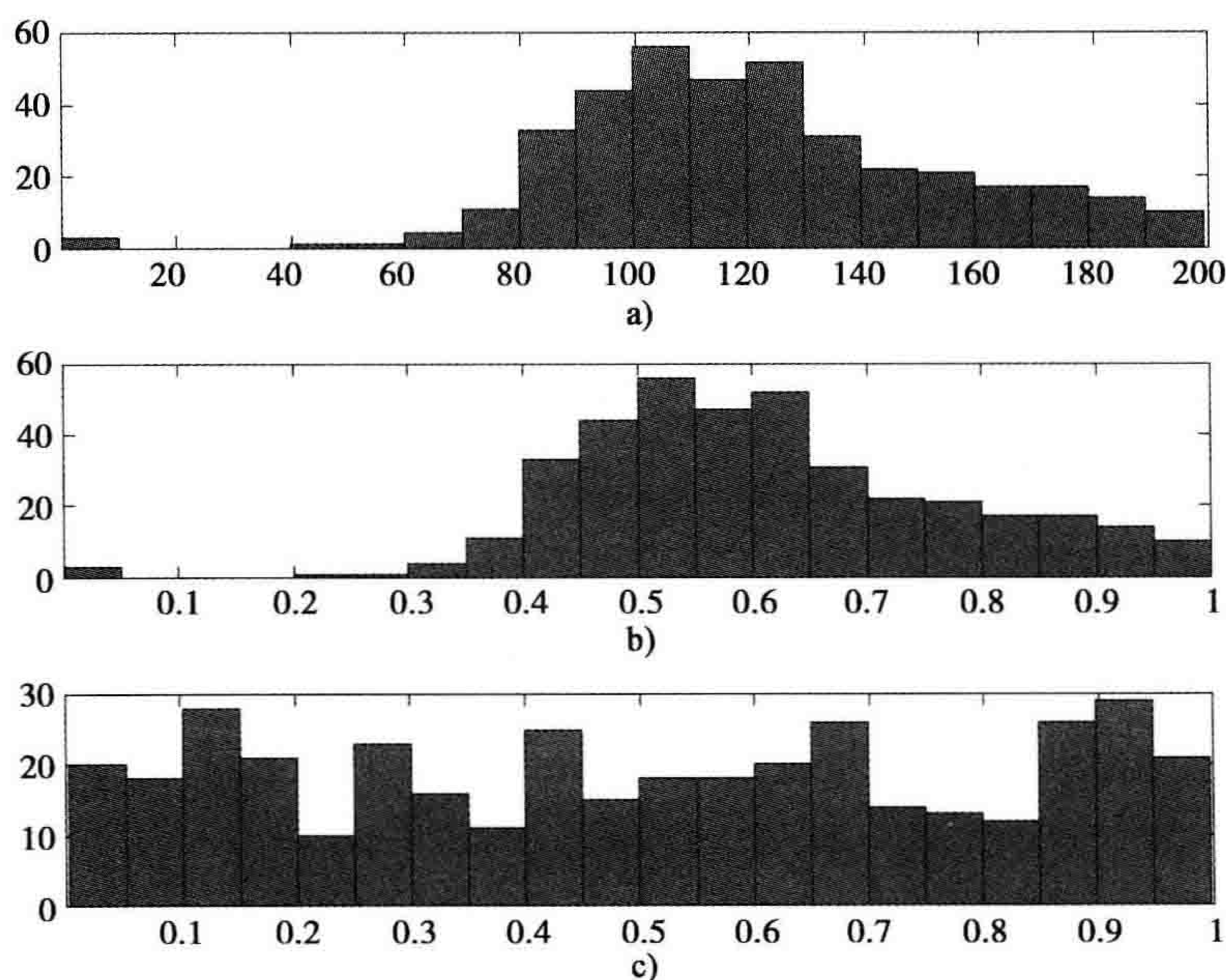


图 3-16 “皮马人-印第安人-糖尿病人”数据集的第二特征向量的直方图

种偏态分布在传统线性规范化之后保持不变(见图 3-16b)，然而，在非线性规范化之后，直方图分布在整个区域[0，1]上，如图 3-16c 所示。

3.6.2 数据集分布

在这个案例研究中，我们使用来自 UCI 机器学习资料库(Asunction & Newman，2007)和 ELENA 工程(Elena，2005)的 19 个数据集。这些数据集的大小和类分布是多样的。表 3-1 描述了这些数据集的主要特征。

表 3-1 数据集特征摘要(按照类偏斜程度排序)

数据集	特征数量	数据数量	少数类样例数量	多数类样例数量	偏斜率
声纳	60	208	97	111	0.47 : 0.53
垃圾电子邮件库	57	4601	1813	2788	0.39 : 0.61
电离层	34	351	126	225	0.36 : 0.64
PID	8	768	268	500	0.35 : 0.65
葡萄酒	13	178	59	119	0.33 : 0.67
德国人	24	1000	300	700	0.30 : 0.70
音素	5	5404	1586	3818	0.29 : 0.71
车辆	18	846	199	647	0.24 : 0.76
纹理	40	5500	1000	4500	0.18 : 0.82
图像分割	18	2310	330	1980	0.14 : 0.86
页面区域	10	5473	560	4913	0.10 : 0.90
卫星图像	36	6435	626	5809	0.10 : 0.90
Mf_Zernike	47	2000	200	1800	0.10 : 0.90
元音	10	990	90	900	0.09 : 0.91
鲍鱼	7	731	42	689	0.06 : 0.94
玻璃	9	214	9	205	0.04 : 0.96
酵母	8	483	20	463	0.04 : 0.96
字母	16	20 000	789	19 211	0.04 : 0.96
航天飞机	9	43 500	37	43 463	0.001 : 0.999

由于其中的几个原始数据集是多类数据，我们对这些数据集进行了一些修改，使它们成为两类问题。表 3-2 给出了修改描述，随后是对每个数据集的简单介绍。

表 3-2 不平衡数据集描述

数据集	少数类	多数类
声纳	“R”类(岩石)	“M”类(金属圆柱体)
垃圾电子邮件库	垃圾邮件	合法邮件
电离层	“坏雷达”类	“好雷达”类
PID	阳性类	阴性类

（续）

数据集	少数类	多数类
葡萄酒	类“1”	类“2”和类“3”
德国人	信用不良的客户	信用良好的客户
音素	“口腔音”类(类1)	“鼻音”类(类0)
车辆	“Van”类	“OPEL”、“SAAS”和“BUS”类
纹理	“13”和“14”类	“2”“3”“4”“6”“7”“8”“9”“10”和“12”类
图像分割	“砖面”类	“天空”“树叶”“水泥墙面”“窗户”“道路”和“草”类
页面区域	“水平线”“图表”、“垂直线”和“图片”类	“文本”类
卫星图像	“潮湿灰色土壤”类	“红色土壤”“棉花作物”“灰色土壤”“带有植被残株的土壤”和“十分潮湿的灰色土壤”类
Mf _ Zernike	数字“9”类	数字“0”“1”“2”“3”“4”“5”“6”“7”和“8”类
元音	类1	类2～11
鲍鱼	“18”类	“9”类
玻璃	类6(餐具)	其他类
酵母	“POX”类	“CYT”类
字母	字母“Z”类	字母“A～Y”类
航天飞机	“Fpv Close”类	“Rad Flow”“Fpv Open”“High”“Bypass”“Bpv Close”和“Bpv Open”类

- 声纳(sonar)：原始数据主要是用来训练一个分类器，以区分反射于金属圆柱体的声纳信号与反射于粗糙的圆柱状岩石的声纳信号。这有两个不同的类标签，“R”(如果样本是岩石)和“M”(如果样本是金属圆柱体)。“R”类设定为少数类，“M”类设定为多数类。
- 垃圾电子邮件库(spambase)：原始数据集被用来分类垃圾电子邮件与合法电子邮件。这个数据库包含4601个电子邮件，其中2788个是合法邮件，1813个是垃圾邮件。每个电子邮件由57个属性表示，其中48个属性编码特定词的频率，6个属性编码特定字符的频率，3个连续属性反映电子邮件中大写字母的统计信息。垃圾邮件和合法邮件分别设定为少数类和多数类。
- 电离层(ionosphere)：原始数据集是用于分类雷达反射信号质量的两类数据集。“坏雷达”的样本作为少数类，“好雷达”的样本作为多数类。
- 皮马人-印第安人-糖尿病人(pima-indians-diabetes，PID)：这是一个两类数据集，用来预测阳性糖尿病人实例。将阳性实例作为少数类，阴性实例作为多数类。
- 葡萄酒(wine)：这个数据集用于通过化学分析确定葡萄酒的来源。原始数据

集包括 3 类：合并类 2 和类 3 作为多数类，类 1 作为少数类。

- 德国人(German)：这是一个两类分类问题，用于德国客户的信用分类。“信用良好的客户”为多数类，“信用不良的客户”为少数类。
- 音素(phoneme)：原始数据集的目的是基于 5 个特征区分鼻音(类 0)和口腔音(类 1)。设定类 0 为多数类，类 1 为少数类。
- 车辆(vehicle)：原始数据集将给定车型分类为 4 种车辆类型之一(OPEL、SAAS、BUS 和 VAN)。设定“VAN”作为少数类，合并剩余的 3 类为多数类(Guo & Viktor，2004b)。
- 纹理(texture)：建立这个数据集的目的是为了用高阶统计量研究纹理区别。数据集共有 11 个类，并具有标签：2、3、4、6、7、8、9、10、12、13、14。合并类 2、3、4、6、7、8、9、10、12 作为多数类，合并剩余类 13 和 14 作为少数类。
- 图像分割(segment)：原始数据集的样本是从包含 7 种户外图像的数据库中随机选取的，包含 7 个类：1(砖面)、2(天空)、3(树叶)、4(水泥墙面)、5(窗户)、6(道路)和 7(草)。因为所有样本的第三个特征是相同的，所以在仿真实验中丢弃此特征。把类 1 作为少数类，合并剩余的其他类作为多数类。
- 页面区域(page block)：这个数据集用于对文档页面布局的所有区域进行分类，共有 5 个类：文本、水平线、图表、垂直线、图片。设定类“文本”作为多数类，合并剩余的 4 个类作为少数类。
- 卫星图像(satimage)：这个数据集用来分类卫星图像中像素在 3×3 邻域内的多光谱值。数据集最初具有 6 个类：1(红色土壤)、2(棉花作物)、3(灰色土壤)、4(潮湿灰色土壤)、5(带有植被残株的土壤)和 6(十分潮湿的灰色土壤)。选择 4 作为少数类，合并剩下的 5 个类作为多数类(Guo & Viktor，2004b)。
- Mf _ Zernike：这个数据集是描述手写数字(0～9)的特征集之一，是从荷兰的一个实用地图中提取出来的。根据文献(Liu 等，2006) 的建议，代表数字 9 的样本作为少数类，剩余的样本作为多数类。
- 元音(vowel)：这是一个分类不同元音的语音识别数据集。原始数据集中有 11 个类。设定类 1 为少数类，合并剩余的类作为多数类(Guo & Viktor，2004b)。
- 鲍鱼(abalone)：这个数据集利用物理测量来预测鲍鱼的年龄。原始数据集中

具有29个类。设定类18为少数类，类9为多数类(Guo & Viktor，2004b)。“性别”特征已经从目前的仿真实验中删除。

- 玻璃(glass)：这个数据集用于根据玻璃的氧化物含量对6种玻璃进行分类。设定类6(餐具)作为少数类，合并剩余的类作为多数类(Guo & Viktor，2004b)。
- 酵母(yeast)：这个数据集用于对蛋白的定位点进行分类。设定类“POX”(过氧化物酶病)作为少数类，类“CYY”(细胞质基质或细胞骨架)作为多数类(Guo & Viktor，2004b)。
- 字母(letter)：建立这个数据集的目的是，从大量黑白矩形像素显示中，确定其中的每一个是26个大写英文字母之一。设定表示字母“Z”的样本是少数类，剩余的为多数类(Liu等，2006)。
- 航天飞机(shuttle)：收集这个数据集的目的是为了探索美国挑战者号航天飞机。特别是，随后调查了航天飞机推进系统的可靠性。最初的7个类是：Rad Flow、Fpv Close、Fpv Open、High、Bypass、Bpv Close和Bpv Open。设定类‘Fpv Close’为少数类，合并剩余的类为多数类。

3.6.3 仿真结果和讨论

在这个案例研究中，所有的仿真结果都是10次运行结果的平均值。在每一次运行中，随机抽出一半数据作为训练数据，剩余的一半作为测试数据。在仿真实验中，用多层感知器神经网络作为基础分类器，具体配置如下：设置隐层神经元的数目为4，输入神经元的数目等于每个数据集的特征的数目。类似于大多数现有的不平衡数据学习方法，在这个案例研究中也考虑两类不平衡问题。因此，在所有的仿真实验中，设定输出神经元的数目为2。Sigmoid函数作为激活函数，设定内部训练次数为100，学习率为0.1。

正如Opitz和Maclin(1999)对集成学习的建议，在当前的仿真实验中用20次自举迭代。在每次自举迭代中，设定所生成的合成数据的数量为少数类样本数量的200%(Chawla等，2002)；对于SMOTEBoost、SMOTE、ADASYN、BorderlineSMOTE和SMOTE-tomek，设定最近邻的数量为5；根据Fan等(1999)的建议，AdaCost的代价因子C设定为3。

表3-3总结了参与比较的算法的性能，其中，突出显示了每个评价指标中性能最佳的算法的指标值。

表 3-3 评价指标和性能比较

数据集	方法	OA	查准率	查全率	*F*-度量	*G*-均值	AUC
声纳	RAMOBoost	**0.7798**	**0.7566**	**0.7813**	**0.7672**	**0.7796**	**0.863 43**
	SMOTEBoost	0.7702	0.7459	0.7748	0.7579	0.7697	0.861 76
	SMOTE	0.7606	0.7330	0.7687	0.7485	0.7605	0.843 11
	ADASYN	0.5712	0.5184	**0.9815**	0.6780	0.4624	0.823 82
	AdaCost	0.7721	0.7559	0.7644	0.7597	0.7711	0.768 64
	BorderlineSMOTE	0.7606	0.7364	0.771	0.7494	0.7607	0.842 05
	SMOTE-tomek	0.7442	0.7379	0.8073	0.7144	0.7448	0.823 78
垃圾电子邮件	RAMOBoost	0.9448	**0.9244**	0.9387	**0.9315**	0.9438	0.983 79
	SMOTEBoost	0.9435	0.9191	0.9418	0.9302	0.9432	0.983 29
	SMOTE	0.9397	0.9194	0.9311	0.9251	0.9382	0.979 42
	ADASYN	0.7746	0.6424	**0.9851**	0.7776	0.7904	0.968 49
	AdaCost	**0.9472**	0.8974	0.9413	0.8588	0.9462	**0.985 52**
	BorderlineSMOTE	0.9291	0.9028	0.936	0.8632	**0.9302**	0.973 62
	SMOTE-tomek	0.9376	0.9002	0.9384	0.8611	0.9377	0.976 49
电离层	RAMOBoost	**0.8411**	**0.8512**	0.6638	**0.744**	**0.7874**	**0.901 38**
	SMOTEBoost	0.8251	0.8244	0.6346	0.7156	0.7662	0.889 07
	SMOTE	0.8177	0.8026	0.6425	0.7106	0.7643	0.820 93
	ADASYN	0.6749	0.5263	**0.7602**	0.6198	0.6912	0.797 78
	AdaCost	0.8337	0.8237	0.6059	0.7352	0.7604	0.881 86
	BorderlineSMOTE	0.8206	0.8466	0.6516	0.7078	0.7698	0.812 65
	SMOTE-tomek	0.8166	0.8494	0.6539	0.7110	0.7677	0.8265
PID	RAMOBoost	0.724	**0.5766**	0.7467	**0.6497**	**0.729**	0.796 08
	SMOTEBoost	0.7229	0.5764	0.74	0.6466	0.7267	0.798 25
	SMOTE	0.7214	0.5746	0.7511	0.6496	0.7281	0.804 28
	ADASYN	0.5539	0.4357	**0.9709**	0.5994	0.5702	0.8144
	AdaCost	**0.7438**	0.2816	0.61	0.3849	0.7043	**0.818 05**
	BorderlineSMOTE	0.7018	0.375	0.7656	0.5029	0.7154	0.7947
	SMOTE-tomek	0.7039	0.3956	0.8102	0.5313	0.7248	0.811 86
葡萄酒	RAMOBoost	**0.9798**	**0.9525**	0.9885	**0.9696**	**0.9813**	**0.999 40**
	SMOTEBoost	0.9787	0.9492	0.9885	0.9678	0.9805	0.999 37
	SMOTE	0.9787	0.9505	0.9885	0.9684	0.9804	0.999 08
	ADASYN	0.7933	0.6094	**1.0000**	0.7536	0.8352	0.996 07
	AdaCost	0.9764	0.9319	0.9813	0.9648	0.9769	0.999 05
	BorderlineSMOTE	0.9753	0.9419	0.9885	0.9681	0.9778	0.997 96
	SMOTE-tomek	0.9551	0.9467	0.9853	0.9696	0.9629	0.997 53
德国人	RAMOBoost	0.7262	**0.5602**	0.5270	**0.5409**	**0.6547**	**0.741 39**
	SMOTEBoost	0.7072	0.5258	0.5126	0.5176	0.6375	0.733 57
	SMOTE	0.6850	0.4878	0.5570	0.5192	0.6420	0.713 65
	ADASYN	0.4918	0.3651	**0.8762**	0.5143	0.5282	0.701 82

（续）

数据集	方法	OA	查准率	查全率	*F*-度量	*G*-均值	AUC
德国人	AdaCost	**0.7482**	0.3963	0.4797	0.5283	0.6446	0.712 54
	BorderlineSMOTE	0.6846	0.4522	0.5754	0.5151	0.6492	0.7105
	SMOTE-tomek	0.691	0.4777	0.6296	0.5148	0.6711	0.734 69
音素	RAMOBoost	0.7921	0.5914	0.9068	**0.7158**	**0.8222**	**0.906 21**
	SMOTEBoost	0.8018	**0.6131**	0.8524	0.7128	0.8159	0.894 72
	SMOTE	0.7860	0.5952	0.8248	0.6899	0.7942	0.871 86
	ADASYN	0.7260	0.5137	**0.9513**	0.6671	0.7770	0.864 97
	AdaCost	**0.8191**	0.2473	0.702	0.3657	0.7797	0.893 95
	BorderlineSMOTE	0.7632	0.3308	0.8741	0.4799	0.7918	0.861 03
	SMOTE-tomek	0.7884	0.2985	0.8151	0.4369	0.7965	0.871 36
车辆	RAMOBoost	0.9655	**0.9142**	0.9398	0.926	0.956	0.994 87
	SMOTEBoost	**0.9667**	0.9137	0.946	**0.929**	0.9591	0.994 46
	SMOTE	0.9589	0.891	0.9373	0.9132	0.9511	0.993 14
	ADASYN	0.821	0.5665	**0.9927**	0.7206	0.8737	0.975 17
	AdaCost	0.9652	0.9132	0.9575	0.371	0.9623	**0.995 11**
	BorderlineSMOTE	0.961	0.9130	0.9652	0.3752	**0.9624**	0.994 05
	SMOTE-tomek	0.9482	0.9091	0.9361	0.3679	0.9436	0.984 98
纹理	RAMOBoost	**0.9991**	**0.9986**	0.9966	**0.9976**	0.9981	**0.999 99**
	SMOTEBoost	0.999	0.9976	**0.997**	0.9973	0.9982	0.999 98
	SMOTE	0.9949	0.9853	0.9863	0.9858	0.9916	0.999 20
	ADASYN	0.9156	0.6837	0.9950	0.8101	0.9453	0.994 87
	AdaCost	0.9987	0.9798	0.9953	0.9946	0.9974	0.999 91
	BorderlineSMOTE	0.9928	0.9783	0.9811	0.9917	0.9881	0.998 56
	SMOTE-tomek	0.9976	0.9793	0.9913	0.9937	0.9951	0.999 67
图像分割	RAMOBoost	**0.9966**	**0.9854**	0.9907	**0.9880**	**0.9941**	0.999 76
	SMOTEBoost	0.9965	0.9853	0.9900	0.9876	0.9938	**0.999 78**
	SMOTE	0.9958	0.9835	0.9863	0.9848	0.9918	0.999 59
	ADASYN	0.9254	0.6253	**1.0000**	0.7980	0.9556	0.999 03
	AdaCost	0.9965	0.9845	0.9913	0.9843	0.9943	0.999 74
	BorderlineSMOTE	0.9954	0.984	0.9869	0.9822	0.9918	0.999 50
	SMOTE-tomek	0.9953	0.984	0.9863	0.9820	0.9915	0.999 61
页面区域	RAMOBoost	**0.9702**	0.8326	0.8928	**0.8614**	**0.9349**	**0.988 99**
	SMOTEBoost	0.9696	**0.8340**	0.8825	0.8573	0.9297	0.987 72
	SMOTE	0.9594	0.7781	0.8563	0.8140	0.9118	0.979 93
	ADASYN	0.9251	0.5862	0.9414	0.7223	0.9322	0.976 21
	AdaCost	0.9704	0.7912	0.8559	0.8469	0.9175	0.988 61
	BorderlineSMOTE	0.9463	0.7853	0.8713	0.8171	0.912	0.970 63
	SMOTE-tomek	0.9576	0.7832	0.8627	0.8168	0.9139	0.977 54
卫星图像	RAMOBoost	0.9195	0.5671	0.7127	**0.6312**	0.819	**0.948 60**

（续）

数据集	方法	OA	查准率	查全率	*F*-度量	*G*-均值	AUC
卫星图像	SMOTEBoost	**0.923**	**0.5867**	0.6717	0.6276	0.7986	0.946 78
	SMOTE	0.8977	0.4791	0.606	0.5327	0.7465	0.897 48
	ADASYN	0.8422	0.3645	0.8431	0.5084	**0.8424**	0.922 34
	AdaCost	0.9217	0.552	0.5426	0.371	0.7118	0.932 55
	BorderlineSMOTE	0.8938	0.685	**0.9652**	0.3752	0.7598	0.901 89
	SMOTE-tomek	0.8957	0.701	0.9361	0.3679	0.773	0.902 51
Mf _ Zermike	RAMOBoost	0.8718	0.3608	0.369	0.3645	0.584	0.894 52
	SMOTEBoost	0.8701	0.3544	0.364	0.3584	0.5798	0.895 37
	SMOTE	0.8838	0.4409	0.592	0.5045	0.7356	0.8922
	ADASYN	0.8634	0.408	**0.809**	**0.5419**	**0.838**	0.906 09
	AdaCost	0.8851	0.3604	0.446	0.3935	0.6441	**0.906 56**
	BorderlineSMOTE	0.8827	0.3678	0.598	0.4217	0.7377	0.8924
	SMOTE-tomek	**0.8877**	**0.3839**	0.745	0.4518	0.8199	0.901 94
元音	RAMOBoost	**0.9988**	**0.9934**	**0.9931**	**0.9931**	**0.9962**	**0.999 90**
	SMOTEBoost	0.9974	0.9842	0.9867	0.9853	0.9925	0.999 88
	SMOTE	0.9794	0.8569	0.9379	0.893	0.9599	0.996 15
	ADASYN	0.9101	0.5095	0.9488	0.6623	0.927	0.985 12
	AdaCost	0.9913	0.903	0.9696	0.9651	0.9813	0.999 06
	BorderlineSMOTE	0.9766	0.8710	0.9222	0.9591	0.9515	0.995 52
	SMOTE-tomek	0.9747	0.8890	0.9382	0.9624	0.9576	0.993 44
鲍鱼	RAMOBoost	0.9405	0.4968	0.4889	0.4813	0.6808	**0.976 09**
	SMOTEBoost	0.943	0.5181	**0.5348**	0.5173	0.7134	0.922 71
	SMOTE	0.9477	**0.5886**	0.5328	0.5412	0.7166	0.922 91
	ADASYN	0.9101	0.361	0.4838	0.3892	0.6684	0.891 79
	AdaCost	**0.9521**	0.241	0.4003	0.455	0.6156	0.923 95
	BorderlineSMOTE	0.9493	0.294	0.4855	**0.554**	0.686	0.903 22
	SMOTE-tomek	0.9441	0.261	0.4319	0.492	0.6433	0.9039
玻璃	RAMOBoost	0.9748	0.6169	0.8464	0.7731	0.8610	0.994 78
	SMOTEBoost	0.9748	0.6480	**0.9464**	0.7430	**0.9596**	0.994 29
	SMOTE	0.9897	**0.8940**	0.9179	**0.8874**	0.9491	**0.998 01**
	ADASYN	0.9421	0.4552	0.7986	0.4970	0.8555	0.977 23
	AdaCost	**0.9907**	0.6377	0.9429	0.7722	0.9625	0.997 41
	BorderlineSMOTE	0.9907	0.6368	0.9262	0.7040	0.9543	0.997 57
	SMOTE-tomek	0.9879	0.6359	0.9119	0.6988	0.9414	0.997 36
酵母	RAMOBoost	0.9581	0.467	0.4341	0.4418	0.6405	0.745 12
	SMOTEBoost	0.9585	0.4941	0.4732	0.4687	0.6651	0.748 78
	SMOTE	0.9722	**0.7557**	**0.5107**	**0.5761**	0.7030	0.816 03
	ADASYN	0.9552	0.5276	0.4891	0.4758	0.6810	0.779 02
	AdaCost	0.9718	0.479	0.4524	0.344	0.6593	0.7792

（续）

数据集	方法	OA	查准率	查全率	F-度量	G-均值	AUC
酵母	BorderlineSMOTE	**0.9726**	0.492	0.4882	0.368	0.6812	0.8096
	SMOTE-tomek	0.9768	0.420	0.5107	0.384	**0.7049**	**0.8241**
字母	RAMOBoost	**0.9982**	**0.9882**	**0.9662**	**0.977**	**0.9827**	**0.999 78**
	SMOTEBoost	0.9977	0.983	0.9591	0.9708	0.979	0.999 77
	SMOTE	0.9921	0.9122	0.8853	0.8981	0.9391	0.995 14
	ADASYN	0.9705	0.5841	0.9122	0.7109	0.942	0.9901
	AdaCost	0.9961	0.9836	0.9261	0.9705	0.9618	0.9989
	BorderlineSMOTE	0.9736	0.9264	0.9003	0.87	0.9375	0.9902
	SMOTE-tomek	0.9925	0.9052	0.8863	0.867	0.9399	0.9943
航天飞机	RAMOBoost	**0.9999**	0.9495	0.9728	**0.9576**	0.9828	**0.9998**
	SMOTEBoost	**0.9999**	0.9442	0.9667	0.9565	0.9828	0.9997
	SMOTE	**0.9999**	**0.9719**	0.9444	0.9538	0.9716	0.9998
	ADASYN	0.9997	0.7984	**1**	0.8855	**0.9999**	0.9995
	AdaCost	**0.9999**	0.9410	0.9667	0.9521	0.9885	0.9998
	BorderlineSMOTE	**0.9999**	0.9400	0.9723	0.9520	0.9857	0.9998
	SMOTE-tomek	**0.9999**	0.9430	0.9333	0.9320	0.9657	0.9998

表3-3的最后一列给出了每种方法的AUC值，并强调了最佳性能。为了展示每个数据集的所有随机运行的ROC曲线，运用Fawcett(2003)中的垂直平均方法生成平均ROC曲线。这种方法的实现可以通过图3-17来说明。假设想要平均两条ROC曲线：l_1 和 l_2，每一条曲线都是由ROC空间中的一系列点形成的。第一步是将fp_rate的范围均匀地划分成一组区间，在每个区间中寻找每条ROC曲线的tp_rate值，并计算其平均值。在图3-17中，X_1 和 Y_1 分别是 l_1 和 l_2 上的点，利用它们可以获得平均ROC曲线上的对应点 Z_1。如果ROC曲线在某个区间上没有对应的点，那么可以在平均ROC曲线上利用线性插值方法获得对应点。例如，在图3-17中，点 $\overline{X}$(对应于fp_rate2)是由两个相邻点 X_2 和 X_3 通过线性插值得到的。一旦获得 $\overline{X}$，就可以与 Y_2 进行平均以获得平均ROC曲线上对应的点 Z_2。在这种方法的基础上，图3-18展示了几条平均ROC曲线，分别对应于数据集"电离层"(见图3-18a)、"音素"(见图3-18b)、"卫星图像"(见图3-18c)和"鲍鱼"(见图3-18d)。

由表3-3所示的仿真结果可以推断，似乎没有哪一种单一的不平衡数据学习算法可以普遍优于所有其他现有的方法。从这些结果还可以观察到几个有趣的现象，例如，除了查全率的性能，似乎ADASYN在这些数据集上能够提供更好的查全率。这是因为ADASYN生成的合成数据样例非常接近决策边界，从而能够更激进地从边界学习(见图3-7c)。这就意味着ADASYN能够迫使算法关注少数(阳性)类数据，

从而不会显著提高其性能。换句话说，如果一个算法将所有测试数据分类为“阳性”（少数类），那么它的“查全”率将会被最大化，即使总体性能很低。这些结果验证了“抽样与自举的集成”小节关于这些算法的不同特点的讨论。

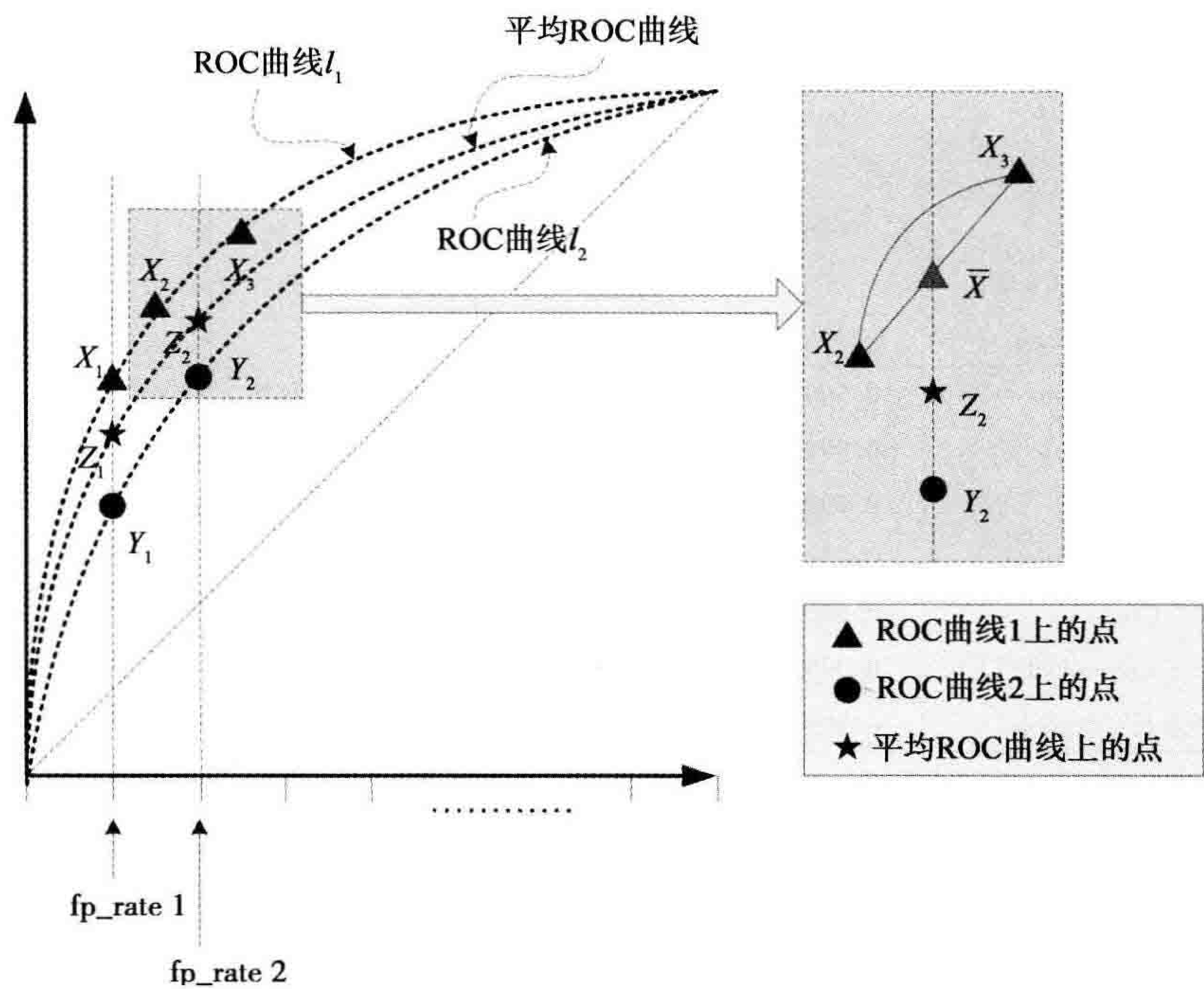
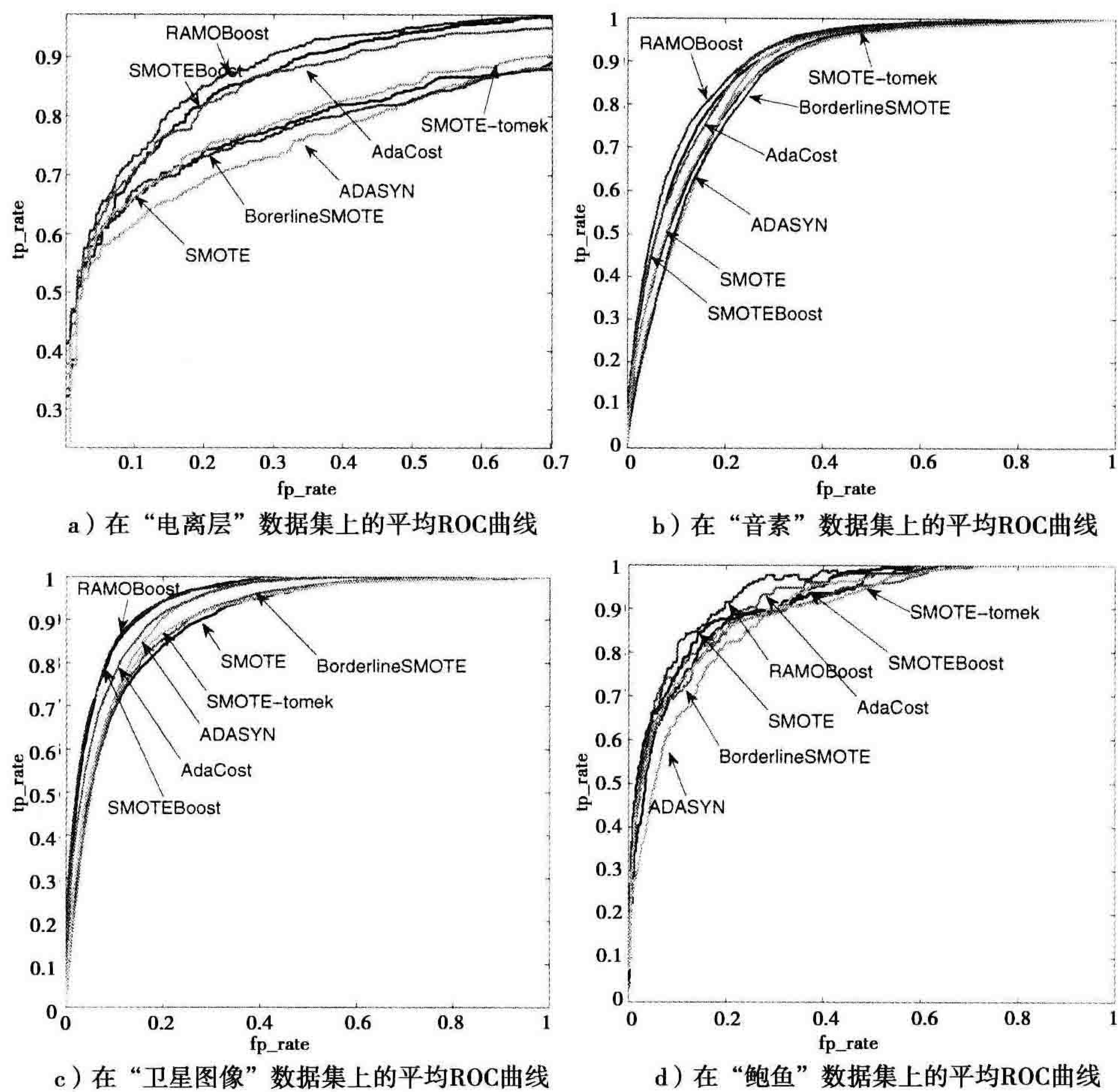


图 3-17　ROC 曲线的垂直平均方法(Fawcett，2003)

当前实例研究中的另一个有趣的实验是：估计不平衡数据学习算法在不同偏斜率下的鲁棒性。这个实验使用“鲍鱼”数据集(28 个类，4177 个样本)作为基准。为了获得各种不平衡率，用不同策略组合原始类，并形成少数类和多数类。表 3.4 总结了组合策略和相应偏斜率的详细信息。这里，采用 AUC 评价指标来评估不平衡数据学习算法在具有不同偏斜率的“鲍鱼”数据集上的性能，如表 3.5 所示。在这种情况下，虽然 RAMOBoost 似乎在性能方面占优势，但是没有任何一种单一的算法可以比其他所有算法表现得更好，即使对于不同偏斜率的相同数据集而言也是如此。结合前面的仿真实验分析，这个结果证明了：没有万能的算法可以处理所有不平衡数据学习问题；同时，多数不平衡数据学习算法对参数配置很敏感。例如，RAMOBoost 和 SMOTEBoost 的迭代次数，ADaCost 的代价因子，甚至一个少数类样本应该用多少个最近邻并由 SMOTE 生成合成样本，所有这些都对算法的性能具有重要的影响。为此，仔细考察不同应用任务中的不平衡数据学习方法的最佳选择和合适的参数配置非常重要。



a）在“电离层”数据集上的平均ROC曲线

b）在“音素”数据集上的平均ROC曲线

c）在“卫星图像”数据集上的平均ROC曲线

d）在“鲍鱼”数据集上的平均ROC曲线

图 3-18 RAMOBoost、SMOTEBoost、SMOTE、ADASYN、AdaCost、BorderlineSMOTE 和 SMOTE-tomek 方法的平均 ROC 曲线

表 3-4 “鲍鱼”数据集中的类组合

序号	少数类组合	多数类组合	少数类编号	多数类编号	偏斜率
I	1⊕2⊕22⊕24⊕25⊕26⊕27⊕28	8⊕9⊕10⊕11	15	2378	0.0063 : 0.9937
II	I⊕23	8⊕9⊕10⊕11	24	2378	0.01 : 0.99
III	I⊕21	8⊕9⊕10⊕11	38	2378	0.0157 : 0.9843
IV	III⊕3	8⊕9⊕10⊕11	53	2378	0.0218 : 0.9782
V	IV⊕20	8⊕9⊕10⊕11	79	2378	0.0322 : 0.9678
VI	V⊕19	8⊕9⊕10⊕11	111	2378	0.0446 : 0.9554
VII	VI⊕18⊕4	8⊕9⊕10⊕11	210	2378	0.0811 : 0.9189
VIII	VII⊕17⊕15	8⊕9⊕10⊕11	371	2378	0.1350 : 0.8650
IX	VIII⊕5	8⊕9⊕10⊕11	486	2378	0.1797 : 0.8303
X	IX⊕6	8⊕9⊕10⊕11	745	2378	0.2386 : 0.7614

表 3-5 不同偏斜率下的 AUC

不平衡率	RAMO-Boost	SMOTE-Boost	SMOTE	ADASYN	Ada-Cost	Broder-lineSMOTE	SMOTE-tomek
0.0063：0.9937	**0.978 87**	0.975 58	0.943 73	0.941 63	0.969 15	0.9166	0.905 82
0.01：0.99	**0.906 48**	0.905 57	0.904 95	0.904 12	0.9049	0.849 15	0.8784
0.0157：0.9843	0.918 86	0.919 13	0.921 22	0.923 61	**0.929 21**	0.914 07	0.894 92
0.0218：0.9782	**0.955 42**	0.950 72	0.933 76	0.932 19	0.951 44	0.896 55	0.929 29
0.0322：0.9678	**0.9584**	0.9523	0.943 71	0.930 93	0.955 62	0.930 95	0.934 87
0.0446：0.9554	**0.944 54**	0.936 52	0.918 17	0.921 62	0.941 09	0.863 02	0.908 76
0.0811：0.9189	0.950 25	0.945 94	0.934 79	0.9348	**0.950 68**	0.929 26	0.913 92
0.1350：0.8650	**0.915 02**	0.905 01	0.899 94	0.872 55	0.911 94	0.895 88	0.8867
0.1797：0.8303	**0.922 74**	0.917 45	0.912 41	0.902 06	0.919 94	0.904 75	0.907 04
0.2386：0.7614	**0.896 96**	0.884	0.875 37	0.877 25	0.893 89	0.869 97	0.869 78

3.7 总结

在本章中，我们讨论了机器智能研究中的不平衡数据学习问题。本章的主要内容包括：

- 不平衡数据学习问题已经成为一个重要的问题，在学术界和工业界吸引了越来越广泛的关注。不平衡数据学习的根本问题在于数据分布和类分布的偏斜性，其目标是开发原理性的方法，在不平衡数据的情况下提高算法的性能。
- 不平衡数据学习问题的本质起源于不平衡分布的数据复杂性，包括不同的概念，如类内不平衡与类间不平衡、内部不平衡与外部不平衡、相对不平衡与稀有样本(绝对稀有)导致的不平衡等。
- 对于抽样法处理不平衡数据学习问题，它是通过特定形式的抽样机制来修改不平衡数据集，以提供平衡的数据分布。然而，这并不意味着基础分类器不能从不平衡数据集学习。代表性的抽样方法包括随机过抽样和欠抽样、告知欠抽样、伴随数据生成的合成抽样、自适应合成抽样、数据清理抽样、基于聚类的抽样和自举集成抽样。
- 代价敏感方法通过使用不同的代价矩阵处理不平衡数据学习问题，其中，代价矩阵描述了错误分类任何特定数据样本的代价。因此，不同于抽样法尝试通过类样本的代表性比例来平衡数据分布，不平衡数据的代价敏感学习方法包括代价敏感数据空间加权(如代价敏感自举抽样)、代价最小化技术(如各种 Meta 技术)和代价敏感拟合技术(如代价敏感决策树和代价敏感神经网络)。
- 基于核的学习方法的原理主要是统计学习和 Vapnik-Chervonenkis(VC) 维度

理论。由于SVM试图最小化总分类误差，因此这类算法天生偏向多数类概念。处理不平衡数据学习的主要核方法包括核方法与抽样技术的集成以及核修改方法。

- 用于不平衡数据学习的主动学习方法在机器学习领域得到了广泛研究。虽然引进主动学习方法是为了解决与未标记训练数据有关的问题，但是越来越多关于不平衡数据集的主动学习问题不断被讨论。例如，基于SVM的主动学习的目的是从未知训练数据中选择富信息的样本(即，最接近超平面的样本)以再次训练核模型。另外，主动学习与抽样技术的集成也得到了一些研究。
- 不平衡数据学习的评价指标对这一领域的研究发展是很重要的。传统的评价指标，如总准确度和总误差率对数据分布的变化高度敏感，因此它们在不平衡数据学习的情况下可能不可靠。为此，在不平衡数据学习领域提出并采用了更多的富信息评价指标，如ROC曲线、PR曲线、代价曲线，以及其他指标。
- 各种实证研究证明，不存在哪种不平衡数据学习技术能适用于所有数据集，并达到一般意义上的“最佳”。因此，对于不同的应用领域，理解数据分布的特性，并对不同的应用提出/采用合适的技术是很重要的。对不平衡数据学习理论的深入理解和利用原则性的方法解决问题，依然是机器学习该领域最具挑战性的问题。

参考文献

Abe, N. (2003). Invited talk: Sampling approaches to learning from imbalanced data sets: Active learning, cost sensitive learning and beyond. *Proc. Int Conf. Machine Learning, Workshop on Learning from Imbalanced Data Sets II*.

Abe, N., Zadrozny, B., & Langford, J. (2004). An iterative method for multiclass cost-sensitive learning. *Proc. ACM SIGKDD Int Conf. Knowledge Discovery and Data Mining*, pp. 3–11.

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced data sets. *Lecture Notes in Computer Science*, *3201*, 39–50.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository [online]. Avaiable: http://www.ics.uci.edu/~mlearn/MLRepository.html*.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20–29.

Borders, Ertekin, S., Weston, J., & Bottou, L. (2005). Fast kernel classifiers with online and active learning. *J. Machine Learning Research*, *6*, 1579–1619.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Chapman & Hall/CRC Press.

Bunescu, R., Ge, R., Kate, R., Marcotte, E., R. Moonet, A. R., & Wong, Y. (2005). Comparative experiments on learning information extractors for protein and their interactions. *Artificial Intelligence in Medicine*, *33*, 139–155.

Caruana, R. (2000). Learning from imbalanced data: Rank metrics and extra tasks. *Proc. Association for the Advancement of Artificial Intelligence Conf.*, pp. 51–57.

Chan, P., & Stolfo, S. (1998). Towards scalable learning with non-uniform class and cost distributions. *Proc. Int Conf. Knowledge Discovery and Data Mining*, pp. 164–168.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, *14*(6), 67–74.

Chawla, N., Japkowicz, N., & Kołcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* (1), 1–6.

Chawla, N. V. (2003). C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree strcuture. *Proc. Int Conf. Machine Learning, Workshop on Learning from Imbalanced Data Sets II*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *J. Artificial Intelligence Research*, *16*, 321–357.

Chawla, N. V., Japkowicz, N., & Kołcz, A. (2003). Workshop on learning from imbalanced data sets II. *Proc. Int. Conf. Machine Learning*.

Chen, K., Lu, B. L., & Kwok, J. (2006). Efficient classification of multilabel and imbalanced data using min-max modular classifiers. *Proc. World Congress on Computation Intelligence - Int Joint Conf. Neural Networks*, pp. 1770–1775.

Chen, S., He, H., & Garcia, E. A. (2010). RAMOBoost: Ranked Minority Oversampling in Boosting. *IEEE Trans. Neural Networks*, *21*, 1624–1642.

Clifton, P., Damminda, A., & Vincent, L. (2004). Minority report in fraud detection: Classification of skewed data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 50–59.

Davis, J., Burnside, E., Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S., & Shavlik, J. (2005). View learning for statistical relational learning: With an application to mammography. *Proc. Int Joint Conf. Artificial Intelligence*, pp. 677–683.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *Proc. Int Conf. Machine Learning*, pp. 233–240.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. *Proc. Int Conf. Knowledge Discoery and Data Mining*, pp. 155–164.

Domingos, P., & Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Proc. Int. Conf. Machine Learning*, pp. 105–112.

Doucette, J., & Heywood, M. (2008). GP classification under imbalanced data sets: Active sub-sampling AUC approximation. *Lecture Notes in Computer Science*, *4971*, 266–277.

Drummond, C., & Holte, R. C. (2000). Exploiting the cost(in)sensitivity of decision tree splitting criteria. *Proc. Int Conf. Machine Learning*, pp. 239–246.

Drummond, C., & Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: Why under sampling beats over-sampling. *Proc. Conf. Machine Learning, Workshop on Learning from Imbalanced Data Sets II*.

Elena Project [Online]. Avaiable: ftp://ftp.dice.ucl.ac.be/pub/neuralnets/elena/databases. (2005).

Elkan, C. (2001). The foundations of cost-sensitive learning. *Proc. Int Joint Conf. Artificial Intelligence*, pp. 973–978.

Ertekin, S., Huang, J., Bottou, L., & Giles, L. (2007a). Learning on the border: Active learning in imbalanced data classification. *Proc. ACM Conf. Information and Knowledge Management*, pp. 127–136.

Ertekin, S., Huang, J., & Giles, C. L. (2007b). Active learning for class imbalance problem. *Proc. Int SIGIR Conf. Research and Development in Information Retrieval*, pp. 823–824.

Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, *20*, 18–36.

Fan, W., Stolfo, S. J., Zhang, J., & Chan, P. K. (1999). AdaCost: Misclassification cost-sensitive boosting. *Proc. Int Conf. Machine Learning*, pp. 97–105.

Fawcett, T. (2001). Using rule sets to maximize roc performance. *Proc. Int Conf. Data Mining*, pp. 131–138.

Fawcett, T. (2003). ROC graphs: Notes and practical considerations for data mining researchers. *Technical Report HPL-2003-4*. (HP Lab)

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proc. Int Conf. Machine Learning*, pp. 148–156.

Freund, Y., & Schapire, R. (2002). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, *55*(1), 119–139.

Fumera, G., & Roli, F. (2002). Support vector machines with embedded reject option. *Proc. Int. Conf. Workshop Pattern Recognition with Support Vector Machines*, pp. 68–82.

Fung, G., & Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning*, *59*(1/2), 77–97.

Gama, J. (2003). Iterative bayes. *Theoretical Computer Science*, *292*(2), 417–430.

Gonzalez, R. C., & Woods, R. E. (2002). Digital image processing. In . (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Guo, H., & Viktor, H. L. (2004a). Boosting with data generation: Improving the classification of hard to learn examples. *Proc. Int. Conf. Innovations Applied Artificial Intelligence*, pp. 1082–1091.

Guo, H., & Viktor, H. L. (2004b). Learning from imbalanced data sets with boosting and data generation: The databoost IM approach. *ACM SIGKDD Explorations Newsletter*, *6*(1), 30–39.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Proc. Int. Intelligent Computing*, pp. 878–887.

Hand, D. J., & Till, R. J. (2001). A simple generalization of the area under the ROC curve to multiple class classification problems. *Machine Learning*, *45*(2), 171–186.

Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proc Int. Conf. Neural Networks*, pp. 1322–1328.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowledge and Data Engineering*, *21*(9), 1263–1284.

He, H., & Shen, X. (2007). A ranked subspace learning method for gene expression data classification. *Proc. Int Conf. Artificial Intelligence*, pp. 358–364.

Holte, R., & Drummond, C. (2005). Cost-sensitive classifier evaluation. *Proc. Int. Workshop Utility-Based Data Mining*, pp. 3–9.

Holte, R. C., Acker, L., & Porter, B. W. (2003). Concept learning and the problem of small disjuncts. *Proc. Int J. Conf. Artificial Intelligence*, pp. 315–354.

Holte, R. C., & Drummond, C. (2000). Explicitly representing expected cost: An alternative to roc representation. *Proc. Int. Conf. Knowledge Discovery and Data Mining*,

pp. 198–207.

Holte, R. C., & Drummond, C. (2006). Cost curves: An improved method for visulization classifier performance. *Machine Learning*, *65*(1), 95–130.

Hong, X., Chen, S., & Harris, C. J. (2008). A kernel-based two-class classifier for imbalanced data sets. *IEEE Trans. Neural Networks*, *18*(1), 28–41.

Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. *Proc. Association for the Advancement of Artificial Intelligence Workshop Learning from Imbalanced Data Sets*, pp. 10–15. (Technical Report WS-00-05).

Japkowicz, N. (2001). Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, *42*, 97–122.

Japkowicz, N. (2003). Class imbalances: Are we focusing on the right issue? *Proc. Int. Conf. Machine Learning. Workshop Learning from Imbalanced Data Sets II*.

Japkowicz, N., Myers, C., & Gluck, M. (1995). A novelty detection approach to classification. *Proc. Joint Conf. Artificial Intelligence*, pp. 518–523.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systemetic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, *6*(1), 40–49.

Joshi, M. V., Kumar, V., & Agarwal, R. C. (2001). Evaluating boosting algorithms to classify rare classes: Comparison and improvements. *Proc. Joint Conf. Data Mining*, pp. 257–264.

Kang, P., & Cho, S. (2006). EUS SVMs: Ensemble of under sampled SVMs for data imbalance problems. *Lecture Notes in Computer Science*, *4232*, 873–846.

Kohavi, R., & Wolpert, D. (1996). Bias plus variance decompostion for zero-one loss functions. *Proc. Int. Conf. Machine Learning*.

Kubar, M. Z., & Kononenko, I. (1998). Cost-sensitive learning with neural networks. *Proc. European Conf. Artificial Intelligence*, pp. 445–449.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, *30*(2/3), 195–215.

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. *Proc. Int. Conf. Machine Learning*, pp. 179–186.

Kwok, J. T. (2003). Moderating the outputs. *Theoretical Computer Science*, *292*(2), 417–430.

Landgrebe, T., Paclik, P., Duin, R., & Bradley, A. P. (2006). Precision-recall operating characteristic (P-ROC) curves in imprecise environments. *Proc. Int. Conf. Pattern Recognition*, pp. 123–177.

Laurikkala, J. (2001). Improving identifications of difficult small classes by balancing class distribution. *Proc. Conf. AI in Medicine in Europe: Artificial Intelligence Medicine*, pp. 63–66.

Learning from imbalanced data sets. (2000). In N. Japkowicz (ed.), *Proc. Association for the Advancement of Artificial Intelligence Workgroup*. (Technical Report WS-00-05).

Lee, H. J., & Cho, S. (2006). The novelty detection approach for difference degress of class imbalance. *Lecture Notes in Computer Science*, *4233*, 21–30.

Li, P., Chan, K. L., & Fang, W. (2006). Hybrid kernel machine ensemble for imbalanced data sets. *Proc. Int. Conf. Pattern Recognition*, pp. 1108–1111.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2006). Exploratory undersampling for class imbalance learning. *Proc. Int. Conf. Data Mining*, pp. 965–969.

Liu, X.-Y., & Zhou, Z.-H. (2006a). The influence of class imbalance on cost-sensitive learning: An empirical study. *Proc. Int. Conf. Data Mining*, pp. 970–974.

Liu, X.-Y., & Zhou, Z.-H. (2006b). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Tran. Knowledge and Data Eng.*, *18*(1), 63–77.

Liu, Y., An, A., & Huang, X. (2006). Boosting prediction accuracy on imbalanced data sets with SVM ensembles. *Lecture Notes in Artificial Intelligence*, *3918*, 107–118.

Liu, Y.-H., & Chen, Y.-T. (2005). Total margin-based adaptive fuzzy support vector machines for multiview face recognition. *Proc. Int. Conf. System, Man, and Cybernetics*, pp. 1704–1711.

Liu, Y. H., & Chen, Y. T. (2007). Face recognition using total margin-based adaptive fuzzy support vector machines. *IEEE Trans. Neural Networks*, *18*(1), 178–192.

Maloof, M., Langley, P., Sage, S., & Binford, T. (1997). Learning to detect rooftops in aerial images. *Proc. Image Understanding Workshop*, pp. 835–845.

Maloof, M. A. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. *Proc. Int. Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II*.

Manevitz, L., & Yousef, M. (2007). One-class document classification via neural networks. *Neurocomputing*, *70*, 1466–1481.

Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. *J. Machine Learning Research*, *2*, 139–154.

McCarthy, K., Zabar, B., & Weiss, G. M. (2005). Does cost-sensitive learning best sampling for classifying rare classes. *Proc. Int. Workshop Utility-Based Data Mining*, pp. 69–77.

Mease, D., Wyner, A. J., & Buja, A. (2007). Boosted classification trees and class probability/quantile estimation. *J. Machine Learning Research*, *8*, 409–439.

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *J. Artificial Intelligence Research*, *11*, 169–198.

Pearson, R., Goney, G., & Shwaber, J. (2003). Imbalanced clustering for microarray time-series. *Proc. Int Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II*.

Platt, J. C. (1999). *Fast Training of Support Vector Machines Using Sequential Minimal Optimization, Advances in Kernel Methods*: Support Vector Learning, pp. 185–208, MIT Press.

Prati, R. C., Batista, G. E. A. P. A., & Monard, M. C. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior. *Proc. Mexican Int. Conf. Artificial Intelligence*, pp. 312–321.

Provost, F. (2000). Machine learning from imbalanced data sets 101. *Proc. Learning from Imbalanced Data Sets: Association for the Advancement of Artificial Intelligence Workshop*. (Technical Report WS-00-05).

Provost, F., & Domingos, P. (2000). Well-trained pets: Improving probability estimation trees. *CeDER Working Paper*: IS-00-04. New York: Stern School of Business, New York University.

Provost, F. J., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. *Proc. Int. Conf. Knowledge Discovery and Data Mining*, pp. 43–48.

Provost, F. J., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparison induction algorithms. *Proc. Int. Conf. Machine Learning*, pp. 445–453.

Qin, A. K., & Suganthan, P. N. (2004). Kernel neural gas algorithms with application to cluster analysis. *Proc. Int. Conf. Pattern Recognition*.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1).

Rao, R. B., Krishnan, S., & Niculescu, R. S. (2006). Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter* (1), 3–10.

Raskutti, B., & Kowalczyk, A. (2004). Extreme re-balancing for SVMs: A case study. *ACM SIGKDD Explorations Newsletter*, *6*(1), 60–69.

Raudys, S. J., & Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Analysis and Machine Learning*, *13*(3), 252–264.

Scholkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*, 1443–1471.

Singla, P., & Domingos, P. (2005). Discriminative training of markov logic networks. *Proc. Nat. Conf. Artificial Intelligence*, pp. 868–873.

Su, C. T., & Hsiao, Y. H. (2007). An evaluation of the robustness of mts for imbalanced data. *IEEE Trans. Knowledge and Data Eng.*, *19*(10), 1321–1332.

Sun, Y., Kamel, M. S., & Wang, Y. (2006). Boosting for learning multiple classes with imbalanced class distribution. *Proc. Int Conf. Data Mining*, pp. 592–602.

Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, *40*(12), 3358–3378.

Taguchi, G., Chowdhury, S., & Wu, Y. (2001). *The mahalanobis-taguchi system*. New York: McGraw-Hill.

Taguchi, G., & Jugulum, R. (2002). *The mahalanobis-taguchi strategy*. Hoboken, NJ: Wiley.

Tan, C., Gilbert, D., & Deville, Y. (2003). multiclass protein fold classification using a new ensemble machine learning approach. *Genome Informatics*, *14*, 206–217.

Tang, Y., & Zhang, Y. Q. (2006). Granular SVM with repetitive undersampling for highly imbalanced protein homology rrediction. *Proc. Int. Conf. Granular Computing*, pp. 457–460.

Tang, Y. C., Jin, B., & Zhang, Y.-Q. (2008). Granular support vector machines with association rules mining for protein homology prediction. *Artificial Intelligence in Medicine special issue on computational intelligence techniques in bioinformatics*, *35*(1/2), 121–134.

Tang, Y. C., Jin, B., Zhang, Y.-Q., Fang, H., & Wang, B. (2005). Granular support vector machines using linear decision hyperplanes for fast medical binary classification. *Proc. Int. Conf. Fuzzy Systems*, pp. 138–142.

Tang, Y. C., Zhang, Y. Q., Huang, Z., Hu, X. T., & Zhao, Y. (2005). Granular SVM-REF feature selection algorithm for reliable cancer-related gene subsets extraction on microarray gene expression data. *Proc. IEEE Symp. Bioinformatics and Bioeng.*, pp. 290–293.

Tashk, A., Bayesteh, R., & Faez, K. (2007). Boosted baysian kernel classifier method for face detectio. *Proc. Int. Conf. Natural Computation*, pp. 533–537.

Ting, K. M. (2000). A comparative study of cost-sensitive boosting algorithms. *Proc. Int. Conf. Machine Learning*, pp. 983–990.

Ting, K. M. (2002). An instance-weighting method to induce cost-sensitive trees. *IEEE Trans. Knowledge and Data Eng.*, *14*(3), 659–665.

Tomek, I. (1976). Two modifications of CNN. *IEEE Trans. System, Man, Cybernetics*, *6*(11), 769–772.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.

Viikki, O., Bye, D., & Laurila, K. (1998). A recursive feature vector normalization

approach for robust speech recognition in noise. *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 733–736.

Vilarino, F., Spyridonos, P., Radeva, P., & Vitria, J. (2005). Experiments with SVM and stratified sampling with an imbalanced problem: Detection of intestinal contractions. *Lecture Notes in Computer Science*, *3687*, 783–791.

Wang, B. X., & Japkowicz, N. (2004). Imbalanced data set learning with synthetic samples. *Proc. IRIS Machine Learning Workshop*.

Wang, B. X., & Japkowicz, N. (2008). Boosting support vector machines for imbalanced data sets. *Lecture Notes in Artificial Intelligence*, *4994*, 38–47.

Webb, G. I., & Pazzani, M. J. (1998). Adjusted probability naive bayesian induction. *Proc. Australian Joint Conf. Artificial Intelligence*, pp. 285–295.

Weiss, G.M. (2005). *Mining Rare Cases, Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, pp. 765–776, Springer.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter*, *6*(1), 7–19.

Weiss, G. M., & Provost, F. (2001). The effect of class distribution on classifier learning: An empirical study. *Technical Report ML-TR-43*. New Brunswick, NJ: Department of Computer Science, Rutgers University.

Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *J. Artificial Intelligence Research*, *19*, 315–354.

Woods, K., Doss, C., Bowyer, K., Solka, J., Priebe, C., & Kegelmeyer, W. (1993). Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *Int J. Pattern Recognition and Artificial Intelligence*, *7*(6), 1417–1436.

Wu, G., & Chang, E. (2003a). Class-boundary alignment for imbalanced data set learning. *Proc. Int. Conf. Data Mining, workshop on Learning from Imbalanced Data Sets II*.

Wu, G., & Chang, E. Y. (2003b). Adaptive feature-space conformal transformation for imbalanced-data learning. *Proc. Int. Conf. Machine Learning*, pp. 816–823.

Wu, G., & Chang, E. (2005). KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. Knowledge and Data Eng.*, *17*(6), 786–795.

Wu, G., & Chang, E. Y. (2004). Aligning boundary in kernel space for learning imbalanced data set. *Proc. Int. Conf. Data Mining*, pp. 265–272.

Yang, W. H., Dai, D. Q., & Yan, H. (2008). Feature extraction uncorrelated discriminant analysis for high-dimensional data. *IEEE Trans. Knowledge and Data Eng.*, *20*(5), 601–614.

Yu, X. P., & Yu, X. G. (2007). Novel text classification based on k-nearest neighbor. *Proc. Int. Conf. Machine Learning Cybernetics*, pp. 3425–3430.

Yuan, J., Li, J., & Zhang, B. (2006). Learning concepts from large scale imbalanced data sets using support vector machines. *Proc. Int. Conf. Multimedia*, pp. 441–450.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *Proc. Int. Conf. Data Mining*, pp. 435–442.

Zhang, J., & Mani, I. (2003). KNN approach to unbalanced data distributions: A case study involving information extraction. *Proc. Int. Conf. Machine Learning, Workshop on Learning from Imbalanced Data Sets*.

Zhou, Z.-H., & Liu, X.-Y. (2006). On multiclass cost-sensitive learning. *Proc. Nat. Conf. Artificial Intelligence*, pp. 567–572.

Zhu, J., & Hovy, E. (2007). Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. *Proc. Joint Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning*,

pp. 783–790.

Zhuang, L., & Dai, H. (2006a). Parameter optimization of kernel-based one-class classifier on imbalance text learning. *Lecture Notes in Artificial Intelligence*, *4099*, 434–443.

Zhuang, L., & Dai, H. (2006b). Parameter estimation of one-class SVM on imbalance text classification. *Lecture Notes in Artificial Intelligence*, *4013*, 538–549.

第4章 集成学习

4.1 引言

集成学习是指开发和集成多个分类器以支持决策过程的机器学习方法。一般来说，集成学习比单个基于模型的学习方法具有更高的精确性和鲁棒性(Kittler, Hatel, Duin & Matas, 1998)，因此在智能计算领域它得到了越来越多的关注。关于集成学习需要解决两个重要问题。首先，如何以一种有原则性的方式设计多个分类器？例如，分类假设的多样性在一个成功的集成学习方法中起着重要作用，因此，如何系统地设计这种多样化的分类器至关重要。目前常用的方法包括 Bagging、AdaBoost、子空间方法、层叠泛化和专家混合体。其次，如何策略性地集成各个分类器的输出，从而得到改进后的最终决策？这个问题通常称为组合投票方法，如几何平均法、算术平均法、中值法和多数投票法等。本章将重点讨论关于集成学习的两个重要问题。

4.2 假设多样性

假设多样性是集成学习的一个公认的关键特性(Kuncheva & Whitaker, 2003; Krogh & Vedelsby, 1995; Rosen, 1996; Lam, 2000; Littlewood & Miller, 1989)。直观地讲，在一个集成学习系统中，如果所有的分类假设的决策相同，那么组合后的分类器与单个分类器做出的决策也相同，因此很难从这些分类器的组合决策中额外受益。于是，理解如何度量集成学习的多样性则至关重要。

假设一个集成学习系统包含 L 个分类假设：$H=\{H_1,H_2,\cdots,H_L\}$，$U=\{x_j,y_j\}$，$(j=1,\cdots,m)$表示有标记的数据集，其中，$x_j\in\mathscr{R}^n$是 n 维特征空间 X 的一个样本，并且 $y_j\in\Omega=\{1,\cdots,C\}$是 x_i 的类别标签。这样，该数据集上每个分类假设 H_i 的输出可以表示为一个 n 维向量：$v_i=[v_{1,i},\cdots,v_{m,i}]$。它满足：如果 H_i 正确预测了 x_j

的类别标签，那么 $v_{j,i}=1$，否则，$v_{j,i}=0$。基于这样的假设，可以用矩阵(类似于混淆矩阵的概念)表示任意两个分类器 H_i 与 H_k 之间的成对关系，如图 4-1 所示。这里，矩阵的每个元素表示分类器的输出关系。例如，m^{11}表示由两种分类器正确分类的样本数目，m^{10}表示 H_i 正确分类但是 H_k 错误分类的样本数目，m^{01}表示 H_k 正确分类但是 H_i 错误分类的样本数目，m^{00}表示被两种分类器都错误分类的样本数目($m=m^{11}+m^{10}+m^{01}+m^{00}$)。

根据图 4-1 的分类器成对表示，Kuncheva & Whitaker(2003)提出了评估多类分类器的标准。

	H_k正确 (表示为“1”)	H_k错误 (表示为“0”)
H_i正确 (表示为“1”)	m^{11}	m^{10}
H_i错误 (表示为“0”)	m^{01}	m^{00}

图 4-1　分类器输出的成对表示

4.2.1　Q 统计量

两个分类器 H_i 和 H_k 的 Q 统计量定义为

$$Q_{i,k}=\frac{m^{11}m^{00}-m^{01}m^{10}}{m^{11}m^{00}+m^{01}m^{10}} \tag{4-1}$$

其中，$Q_{i,k}\in[-1,1]$。对于在统计上独立的分类器(最大多样性)，$Q_{i,k}$的期望为 0。对于 L 个分类器，所有分类器对的 Q 统计量的平均值定义为

$$Q_{av}=\frac{2}{L(L-1)}\sum_{i=1}^{L-1}\sum_{k=i+1}^{L}Q_{i,k} \tag{4-2}$$

4.2.2　相关系数

在这种情况下，多样性由两个分类器输出的相关性来衡量：

$$\rho_{i,k}=\frac{m^{11}m^{00}-m^{01}m^{10}}{\sqrt{(m^{11}+m^{10})(m^{01}+m^{00})(m^{11}+m^{01})(m^{10}+m^{00})}} \tag{4-3}$$

一般来说，Q 统计量和相关系数 ρ 有相同的正负号，且满足 $|\rho|\leqslant|Q|$(Kuncheva & Whitaker，2003；Kuncheva，Whitaker，Shipp & Duin，2003)。与 Q 统计量相似，当 $\rho=0$ 时，多样性达到最大，这意味着两个分类器是不相关的。

4.2.3 不一致度量

不一致度量被定义为两个分类器不一致的样本数目与总样本数目的比率：

$$\xi_{i,k}=\frac{m^{01}+m^{10}}{m^{11}+m^{10}+m^{01}+m^{00}} \tag{4-4}$$

直观上，不一致度量反映了两个分类器之间的不一致性，并且也可以用来度量多样性。例如，在 Skalak(1996)中不一致度量用来分析一个基分类器和一个互补分类器之间的多样性；在 Ho(1998b)中，不一致度量也用于分析决策森林方法的多样性。

4.2.4 双错度量

双错度量定义为被两个分类器错误分类的样本数目占总样本数目的比率(Giacinto & Roli，01)。

$$\gamma_{i,k}=\frac{m^{00}}{m^{11}+m^{10}+m^{01}+m^{00}} \tag{4-5}$$

双错度量在许多文献中用于评估集成学习的多样性，如 Giacinto、Roli(2001)以及 Yang、Wang 和 He(2007)。

4.2.5 熵度量

除了上述的成对度量之外，还有非成对的多样性度量，首先讨论熵度量 E。假设：如果有一半分类器是正确分类的，而剩下的一半是错误分类的，则表示假设多样性程度最高。假设 δ_j 表示 L 个分类器中正确分类了样本 x_j 的分类器的数目，则熵度量定义为

$$E=\frac{1}{m}\sum_{j=1}^{m}\frac{1}{(L-\lceil L/2\rceil)}\min(\delta_j,L-\delta_j) \tag{4-6}$$

其中，$\lceil\cdot\rceil$ 是下取整算子。熵度量在 0～1 之间变化，0 表示无多样性，1 表示多样性程度最高(Kuncheva & Whitaker，2003)。

4.2.6 Kohavi-Wolpert 方差

Kohavi-Wolpert(KW)方差(Kohavi & Wolpert，1996)被定义为

$$\mathrm{KW}=\frac{1}{mL^2}\sum_{j=1}^{m}\delta_j(L-\delta_j) \tag{4-7}$$

KW 多样性度量与式(4-4)所示的不一致度量密切相关。事实上，可以证明，

KW 度量是不一致度量的标准化版本(Kuncheva & Whitaker，2003)：

$$\mathrm{KW}=\frac{L-1}{2L}\xi_{\mathrm{av}} \tag{4-8}$$

其中，对于所有的分类器对，ξ_{av}是不一致度量 $\xi_{i,j}$ 的均值，与式(4-2)的计算类似。

4.2.7 测试者间的一致性

测试者间的一致性 κ 度量与评价者间信度密切相关，该信度用来评估一致性水平、同类相关系数和 Looney 的重要性测试(Kuncheva & Whitaker，2003；Fleiss，1981；Looney 1988；Dietterich，2000)。特别地，$\bar{p}$ 表示单个分类器的平均分类精度：

$$\bar{p}=\frac{1}{mL}\sum_{j=1}^{m}\sum_{i=1}^{L}v_{j,i} \tag{4-9}$$

测试者间的一致性 κ 度量定义为

$$\kappa=1-\frac{(1/L)\sum_{j=1}^{m}\delta_j(L-\delta_j)}{m(L-1)\bar{p}(1-\bar{p})} \tag{4-10}$$

从式(4-4)和式(4-7)可知，测试者间的一致性 κ 度量也与 KW、ξ_{av} 相关(Kuncheva & Whitaker，2003)：

$$\kappa=1-\frac{L}{(L-1)\bar{p}(1-\bar{p})}\mathrm{KW}=1-\frac{1}{2\bar{p}(1-\bar{p})}\xi_{\mathrm{av}} \tag{4-11}$$

4.2.8 困难程度

困难程度的思想最早出现在 Hansen 和 Salamon(1990)中，用于神经网络集成研究。为了定义困难程度，首先定义一个随机变量 $Z(x_j)=\left\{\frac{0}{L},\frac{1}{L},\cdots,\frac{L}{L}\right\}$，它表示 H 中正确分类随机选取的样本 x_j 的分类器的比例。这样，困难程度 θ 被定义为随机变量 Z 的方差(Kuncheva & Whitaker，2003；Hansen & Salamon，1990；Polikar，2006)：

$$\theta=\frac{1}{L}\sum_{t=1}^{L}(z_t-\bar{z})^2 \tag{4-12}$$

其中，$\bar{z}$ 为 z 的均值，θ 度量的关键思想基于困难程度的分布。例如，在极端情况下，如果所有的分类器相同，Z 分布将变为两个序列(在概率质量函数空间中，一个为 0，另一个为 1)(Kuncheva & Whitaker，2003)，较大的 θ 值表示弱多样性。然而，如果分类器是负相关的，也就是说，一些样本对于 H 中的一些分类器是困难的，而对另一些分类器是简单的，那么 θ 相对较小，这表示强多样性。

4.2.9 广义多样性

广义多样性的概念最早是由 Partridge 和 Krzanowski(1997)提出的。广义多样性的基础是如下假设：从 H 中随机选择两个分类器，其中一个分类失败是伴随着另一个分类正确发生的，此时多样性程度最高(Kuncheva & Whitaker，2003；Partridge & Krzanowski，1997)。为此，令随机变量 Y 表示对随机选取的样本 x_j 错误分类的分类器在 H(总数为 L 个分类器)中的比例(也就是，$Y=1-Z$)，定义 p_i 为 $Y=\frac{i}{L}$ 的概率，也就是说，p_i 为 i 个随机选取的分类器在一个随机选取的样本 x_j 上错误分类的概率。这样，广义多样性(GD)可定义为

$$\mathrm{GD}=1-\frac{\sum_{i=1}^{L}\frac{i(i-1)}{L(L-1)}p_i}{\sum_{i=1}^{L}\frac{i}{L}p_i} \tag{4-13}$$

其中，GD 的值在 0～1 变化，GD=0 表示多样性程度最低，GD=1 表示多样性程度最高(Partridge & Krzanowski，1997)。Partridge & Krzanowski(1997)讨论了对 GD 概念的改进，即一致失败多样性(CFD)。

总之，本节主要讨论了分类器多样性度量方法，如表 4-1 所示。这里“+”和“－”符号表示的多样性变化趋势是随着相对应的度量变化而变化的。例如，随着每一行中对应度量的增加，“+”表示多样性程度增强，而“－”表示多样性程度减弱。

表 4-1 分类器多样性度量总结(Kuncheva & Whitaker，2003)

度量	符号	+/−	成对(是或否)	对称(是或否)	参考
Q统计量	Q	−	是	是	Yule(1900)
相关系数	ρ	−	是	是	Sneath & Sokal(1973)
不一致度量	$\xi_{i,j}$	+	是	是	Skalak(1996)，Ho(1998b)
双错度量	γ	−	是	否	Giacinto and Roli(2001)
熵度量	E	+	否	是	Cunningham and Carney(2000)
Kohavi-Wolpert 方差	KW	+	否	是	Kohavi and Wolpert(1996)
测试者间的一致性	κ	−	否	是	Fleiss(1981)，Looney(1988)，Dietterich(2000)
困难度量	θ	−	否	否	Hansen & Salamon(1990)
广义多样性/一致失败多样性	GD/CFD	+	否	否	Partridge & Krzanowski(1997)

4.3 多分类器的研究进展

多样化分类器的集成是集成学习的第一步，实现这一目标有多种方法。本节主要讨论以下几种方法。

4.3.1 自举聚集

自举聚集(Bagging)最早由 Breiman(1996)提出。Bagging 的关键思想是在样本空间中采用自举放回抽样来创建多个分类器。在 Bagging 方法中，一个分类器的训练集通过均匀抽样函数从整个训练样本集中抽样得到，然后用一个简单的投票方案组合预设的有限数目的分类器来预测测试样本。简单地说，Bagging 算法可以归纳如下(Breiman，1996)。

[算法 4.1] Bagging 算法

输入：

- 设训练集 U 包含 m 个样本，可表示为 $\{\boldsymbol{x}_j, y_j\}, (j=1,\cdots,m)$，其中 $\boldsymbol{x}_j$ 是 n 维特征空间 X 的一个样本，且 $y_j \in \Omega=\{1,\cdots,C\}$ 是 $\boldsymbol{x}_j$ 的类别标签；
- 基本学习算法：WeakLearn；
- 整数 T 表示迭代次数。

设 $t=1$，…，L，则

1)通过随机抽取 m 个样本，获得一个自举样本 S_t，并更换原来的训练集 $\boldsymbol{U}$。

2)基于 S_t 调用 WeakLearn，以设计分类器 h_t。

测试阶段：

用多数投票法组合 L 个分类器 $H=\{H_1,\cdots,H_L\}$ 的输出，预测每个测试样本的最终类别标签。

4.3.2 自适应增强

在 Bagging 中，每一个样本被赋予相同的权重。因此，那些难于学习的样本与易于学习的样本之间并无区别。为了解决这一问题，各种各样的 Boosting 算法被提出，以根据样本的分布和对其学习的困难程度自适应地调整权重。

最流行的自举算法是自适应增强(AdaBoost)算法的变形 AdaBoost. M1 和 AdaBoost. M2(Freund & Schapire，1996，1997)。AdaBoost 算法的基本思想是，根据

训练样本的学习能力迭代更新其权重(Freund & Schapire，1996，1997)。AdaBoost 算法通过对训练数据迭代地应用一个自举分布 D_t 来实现集成学习。这里，根据具体实现，分布 D_t 可以被表示为不同的形式。例如，对于基本学习算法(WeakLearn)，可以直接加权，并把这些权重直接应用到 Boosting 算法中(二次加权)。另一方面，如果 WeakLearn 需要未加权的训练样本，根据自举分布 D_t(重抽样)，可以从训练数据中随机选择(更换)一组样本。在每次迭代中，AdaBoost 算法自适应地改变 D_t，使得难于学习的样本的权重大于易于学习的样本的权重(Freund & Schapire，1996，1997)。这样，决策边界自适应地偏向那些难于学习的样本。Freund & Schapire(1997) 给出了关于 Boosting 算法误差率的界的大量理论分析，Freund & Schapire，(1996) 和 Opitz & Maclin(1999) 在不同领域的实验中展示了该方法的强大功能。

[算法 4.2] AdaBoost. M1

输入：

- 训练数据集 U 包含 m 个样本，可表示为 $\{x_j, y_j\}$，$(j=1,\cdots,m)$，其中 $\boldsymbol{x}_j$ 是 n 维特征空间 X 的一个样本，$y_j \in \Omega = \{1, \cdots, C\}$ 是 $\boldsymbol{x}_j$ 的类别标签；
- 一个基本学习算法：WeakLearn；
- 整数 T 表示迭代次数。

初始化：

$D_1(j)=1/m, j=1,\cdots,m$

设 $t=1,\cdots,L$，则

1)基于分布 D_t 调用 WeakLearn 并设计分类器 h_t；

2)设计分类器 h_t：$X \to \Omega$；

3)计算 h_t 的误差率：

$$\varepsilon_t = \sum_{f:h_t(x_j) \neq y_j} D_t(j) \tag{4-14}$$

如果 $\varepsilon_t > 1/2$，则 $L=t-1$ 且迭代终止。

4)令 $\beta_t = \dfrac{\varepsilon_t}{1-\varepsilon_t}$；

5)更新 D_t 的分布：

$$D_{t+1}(j) = \frac{D_t(j)}{Z_t} \times \begin{cases} \beta_t, & h_t(x_j) = y_j \\ 1, & 其他 \end{cases} \tag{4-15}$$

其中，Z_t 是归一化常数，D_{t+1} 为分布函数($\sum D_{t+1}=1$)。

输出：

最终分类器如下

$$h_{\text{final}}(x)=\arg\max_{y\in\Omega}\sum_{t:h_t(x)=y}\lg\frac{1}{\beta_t} \tag{4-16}$$

Freund 和 Schapire(1997)给出了 AdaBoost. M1 的重要理论分析。特别地，在理论上证明了：如果 WeakLearn 能够持续保持略小于 0.5 的误差率(对于二元分类情况，意味着 WeakLearn 略好于随机猜测)，那么最终分类器 h_{final} 的训练误差率以指数形式快速逼近 0，则可以表示为如下形式(Freund & Schapire，1997)。

定理 1(Freund & Schapire，1997)：假设 AdaBoost. M1 产生一系列误差率为 ε_t $(t=1,\cdots,L)$ 的分类器 $h_t,t=1,\cdots,L$，每个误差率 $\varepsilon_t\leqslant 1/2$，定义 $r_t=\frac{1}{2}-\varepsilon_t$，则最终分类器 h_{final} 的误差率上界如下：

$$\frac{1}{m}\left|\{j:h_{\text{final}}(x_j)\neq y_j\}\right|\leqslant\prod_{t=1}^{L}\sqrt{1-4r_t^2}\leqslant\exp\left(-2\sum_{t=1}^{L}r_t^2\right) \tag{4-17}$$

AdaBoost. M1 算法的局限性是要求每个 WeakLearn 的误差率小于 0.5。对于多类分类问题，这种要求较高，基本的 WeakLearn 也许不能满足。为了克服这一局限性，Freund 和 Schapire 将 AdaBoost. M1 算法扩展为 AdaBoost. M2 算法。AdaBoost. M2 算法的核心思想是：用伪损失概念代替 AdaBoost. M1 算法中的单标签表示，以实现信息量更丰富的误差率度量。特别地，伪损失是通过所有样本对及其错误标签的分布来计算的。这样 AdaBoost. M2 不仅关注难于分类的样本，更重要的是，它还能关注难以区分的错误标签(Freund & Schapire，1997)，AdaBoost. M2 算法概括如下(Freund & Schapire，1996，1997)。

[算法 4.3] AdaBoost. M2

输入：

- 训练数据集 U 包含 m 个样本，可表示为 $\{x_j,y_j\},(j=1,\cdots,m)$，其中 x_j 是 n 维特征空间 X 的一个样本，$y_j\in\Omega=\{1,\cdots,C\}$ 是 x_j 的类别标签；
- 一个基本学习算法：WeakLearn；
- 整数 L 表示迭代次数。

定义 B 为所有错误标签样本集：

$$B=\{(j,y):j\in\{1,\cdots,m\},y\neq y_j\} \tag{4-18}$$

初始化：

对于$(j,y)\in B$，$D_1(j)=1/|B|$，其中$|B|$表示B的大小。

设$t=1,\cdots,L$，则

1)基于分布D_t调用WeakLearn，开发分类器$h_t-X\times\Omega\rightarrow[0,1]$；

2)计算h_t的误差率：

$$\varepsilon_t=\frac{1}{2}\sum_{(j,y)\in B}D_t(j,y)(1-h_t(x_j,y_j)+h_t(x_j,y)) \tag{4-19}$$

3)令$\beta_t=\frac{\varepsilon_t}{1-\varepsilon_t}$；

4)更新分布函数D_t：

$$D_{t+1}(j,y)=\frac{D_t(j,y)}{Z_t}\cdot\beta_t^{(1/2)(1+h_t(x_j,y_j)-h_t(x_j,y))} \tag{4-20}$$

其中，Z_t是归一化常数，D_{t+1}为分布函数($\sum D_{t+1}=1$)。

输出：

最终分类器如下

$$h_{\text{final}}(x)=\arg\max_{y\in\Omega}\sum_{t:h_t(x)=y}\left(\lg\frac{1}{\beta_t}\right)h_t(x,y) \tag{4-21}$$

与AdaBoost.M1算法相比，AdaBoost.M2算法在实现方面的重要性在于用错误标签表示原始数据集。尤其是，一个错误标签是一对(j,y)，其中j是训练样本的索引，y是样本j的一个错误标签(见式(4-18))，并且错误标签分布是定义在所有错误标签集B上的分布。每一次自举迭代对WeakLearn输入一个错误标签分布D_t，下面的例子给出了更清楚的说明。

例如：考虑一个三类($C=3$)分类问题，$\Omega=\{1,2,3\}$，具有5个样本($m=5$)，相关的标签类别为$(x_1,1)$、$(x_2,1)$、$(x_3,2)$、$(x_4,3)$和$(x_5,3)$。图4-2给出了AdaBoost.M1算法和AdaBoost.M2算法初始化过程中的D_1分布。从图中可以清楚地看到AdaBoost.M1算法和AdaBoost.M2算法的初始分布D_1分别设为$\frac{1}{m}$和$\frac{1}{m(C-1)}$。一旦设置了初始分布，自举过程可以按照上述算法进行。

AdaBoost.M2算法的目标是寻找具有式(4-19)定义的较小伪损失的弱分类器h_t。这对AdaBoost.M2算法的实际应用提出了另一个具体要求，要求基础学习算法能够对每个数据样本潜在的类别标签产生概率值：$X\times\Omega\rightarrow[0,1]$。因此，可能需要对标准“现成的”基础学习算法进行修改，但是通常这种修改是简单的(Freund &

Schapire，1996）。与 AdaBoost. M1 相似，关于最终分类器，Freund 和 Schapire (1997)也给出了理论上训练误差率的边界。

原始数据样本
三类分类问题 (C=3)
5个样本 (m=5)

数据样本	类标签
x_1	1
x_2	1
x_3	2
x_4	2
x_5	3

AdaBoost.M1

j	x_j	$D_1(j)$
j=1	x_1	1/5
j=2	x_2	1/5
j=3	x_3	1/5
j=4	x_4	1/5
j=5	x_5	1/5

$D_1(j)=1/m$

AdaBoost.M2

B	(x_j,y)	$D_1(j,y)$
(j=1,y=2)	$(x_1,2)$	1/10
(j=1,y=3)	$(x_1,3)$	1/10
(j=2,y=2)	$(x_2,2)$	1/10
(j=2,y=3)	$(x_2,3)$	1/10
(j=3,y=1)	$(x_3,1)$	1/10
(j=3,y=3)	$(x_3,3)$	1/10
(j=4,y=1)	$(x_4,1)$	1/10
(j=4,y=2)	$(x_4,2)$	1/10
(j=5,y=1)	$(x_5,1)$	1/10
(j=5,y=2)	$(x_5,2)$	1/10

$D_1(j,y)=1/(m(C-1))$

图 4-2 AdaBoost. M1 和 AdaBoost. M2 的初始化分布

定理 2(Freund & Schapire，1997)：假设 AdaBoost. M2 算法生成了一系列伪损失为 ε_t 的分类器 h_t，$t=1$，…，L，定义 $r_t=\frac{1}{2}-\varepsilon_t$，则最终分类器 h_{final} 的误差率上界如下：

$$\frac{1}{m}\left|\{j:h_{\text{final}}(x_j)\neq y_j\}\right|\leqslant(C-1)\prod_{t=1}^{L}\sqrt{1-4r_t^2}$$

$$\leqslant(C-1)\exp\left(-2\sum_{t=1}^{L}r_t^2\right) \tag{4-22}$$

其中，C 为类别数量。

4.3.3 子空间方法

Bagging 和 AdaBoost 均属于在样本空间中设计多分类器的方法。另一类主要方法是在特征空间中设计多分类器。本节主要讨论随机子空间方法和排序子空间方法。

随机子空间方法的关键是通过随机选择特征空间的子集来设计多分类器。例如，决策森林(Ho，1998a，1998b，1995)和随机森林(Breiman，2001)，通过伪随机选择的特征空间的子集系统地构建多分类器。例如，Ho讨论了对于一个给定的 n 维特征空间可以有 2^n 种选择，其中每个选择可用于构造决策树(Ho，1998b)。此外，如果决策树的内部子空间发生了变化(分离出不同的特征维)，则会很容易构造更多不同的树。决策森林也是一种并行学习算法，其中的每个决策树都是独立的(Ho，1998b)。现有的许多研究表明，随机子空间方法为不同领域的应用提供了具有竞争力的学习性能(Ho，1998a，1998b，1995；Breiman，2001)。

除了随机地选择子空间特征设计多分类器之外，He和Shen(2007)提出了针对集成学习的排序子空间方法(RS)。RS方法的主要思想是根据特征的评分函数对所有特征进行非升序排序，以促进多分类器的设计。这样，可以对不同权重的特征空间根据其重要性进行自举采样来设计多分类器。与决策森林和随机森林方法的随机特征子空间选择过程不同，RS方法使用抽样概率函数，用更系统的方法从大量信息中选取特征子空间。RS方法概括如下。

[算法 4.4] 排序子空间

输入：

- 训练集 U 具有 m 个样本，可表示为 $\{\boldsymbol{x}_j, y_j\}, (j=1,\cdots,m)$，其中 $\boldsymbol{x}_j$ 是 n 维特征空间 X 的一个样本，并且 $y_j \in \Omega=\{1,\cdots,C\}$，是与 $\boldsymbol{x}_j$ 有关的类别标签；
- 一个基本学习算法：WeakLearn；
- 整数 L 表示迭代次数。

1)定义一个特征评分函数 $S(\cdot)$，并且计算每一个特征的得分 $S_f(i), i=1,\cdots,n$；

2)根据特征得分，对所有特征进行非升序排列：

$$S_f(i):\{S_f(1),S_f(2),\cdots,S_f(n)\} \quad 其中, S_f(1)\geqslant S_f(2)\geqslant\cdots\geqslant S_f(n)$$

3)根据特征得分值计算特征采样分布函数：$D(i)=\dfrac{S_f(i)}{Z}$，其中，Z 是归一化常数，$D(i)$ 是一个分布，满足 $\sum D(i)=1$。

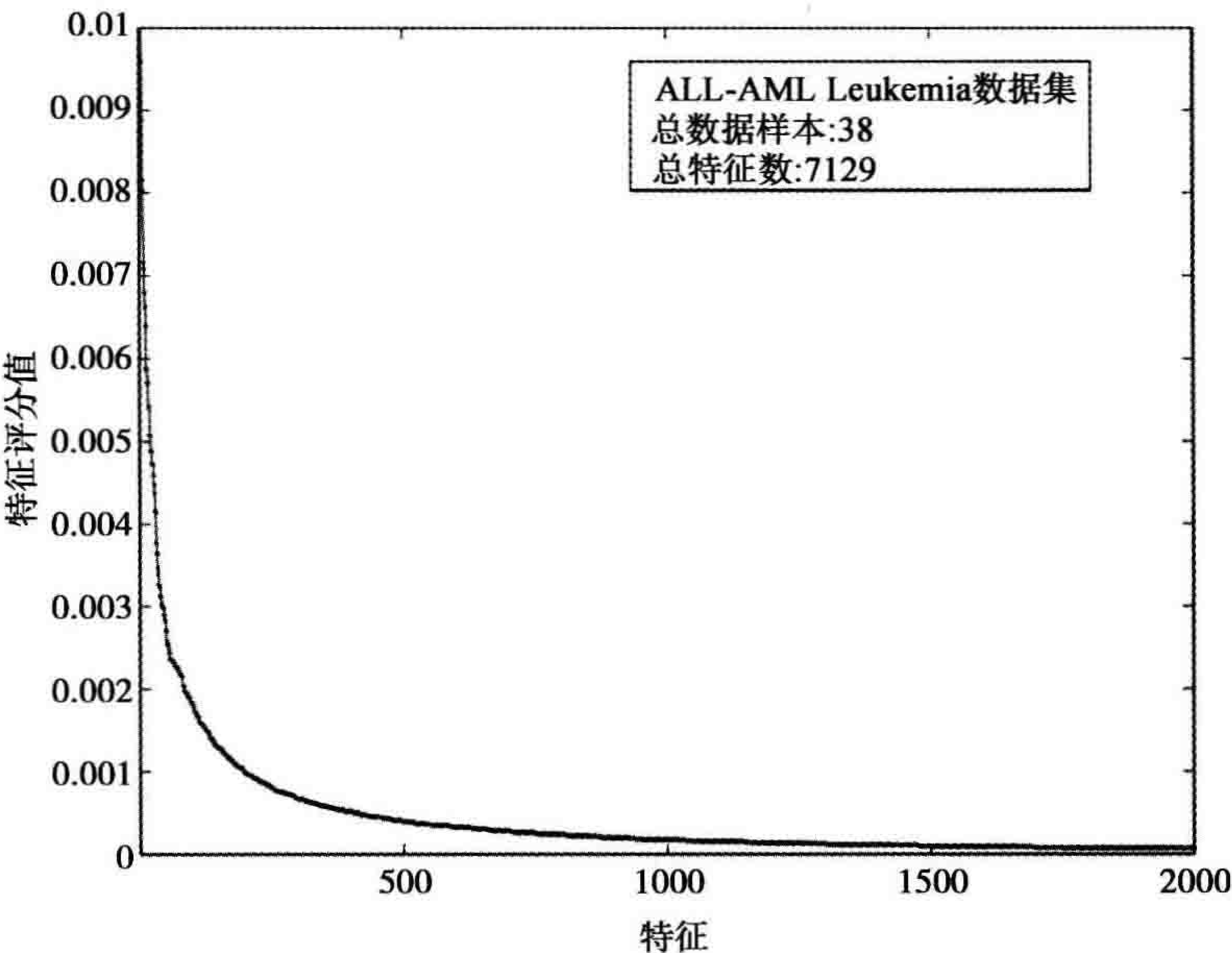
设 $t=1$，…，L，则

1)在分布为 $D(i)$ 的特征空间中进行自举采样(有放回的)，特征子空间表示为 F_t；

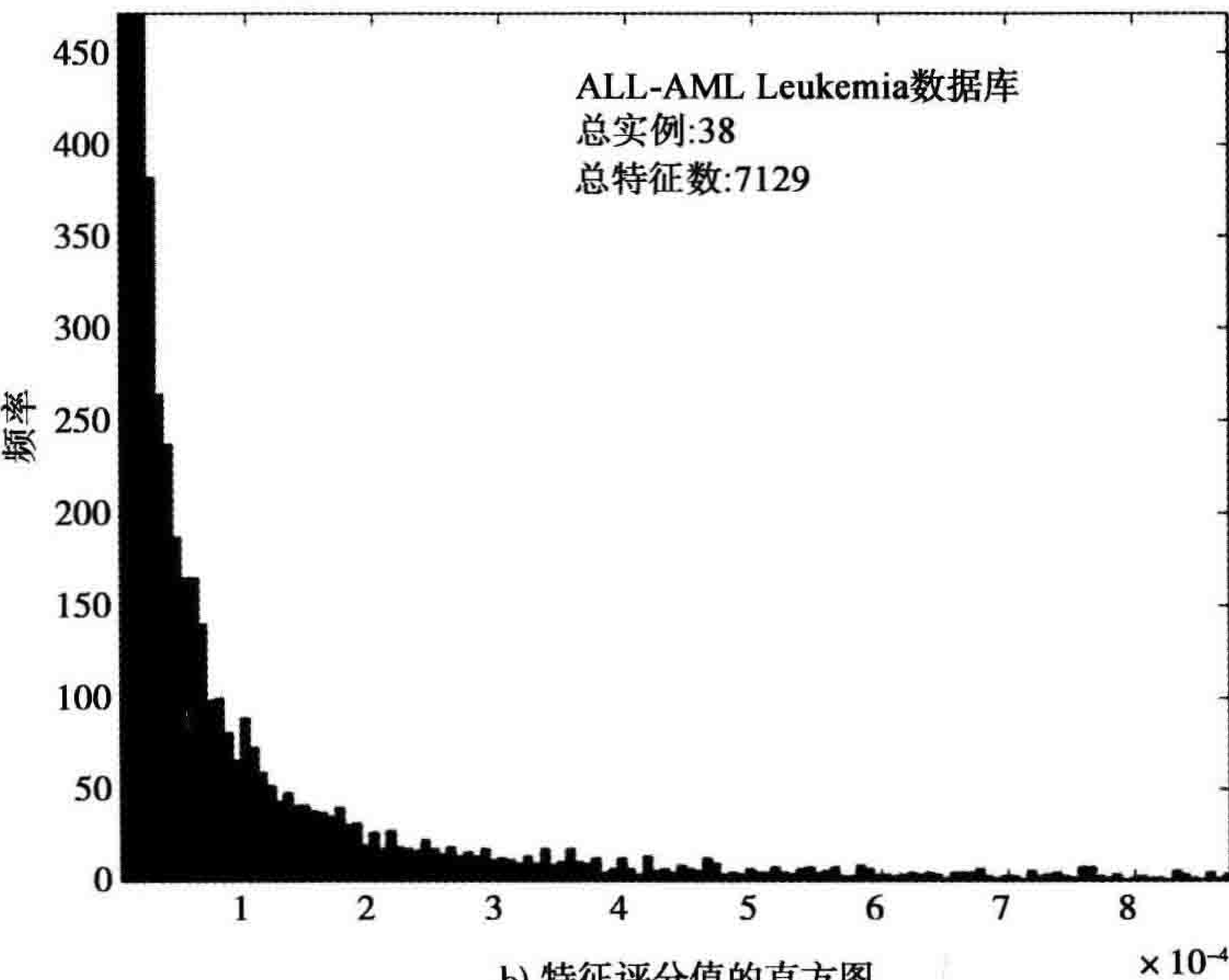
2)将特征空间 F_t 归一化为 F_{ut}；

3)基于分布 D_t 调用 WeakLearn，设计分类器 h_t：$X\rightarrow\Omega$，其中 D_t 具有特征子空间 F_{ut}。

RS算法的输出是一组多分类器，可以与不同的组合投票方法相结合，详见4.4节。在具体实现RS算法时，要用到一个特征评分函数，该函数可以根据不同的应用计算获得。例如，He和Shen(2007)用一个与Fisher判别准则相关的特征评分函数分析基因数据。图4-3a和b展示了排序特征评分值及其对应的直方图。



a) 微阵列数据分析的特征评分值（放大表示）



b) 特征评分值的直方图

图4-3 RS方法的特征评分分布

4.3.4 层叠泛化

层叠泛化是集成学习的另一种强有力的技术(Wolpert，1992)。层叠泛化的关键思想是设计多层次分类器(通常是两层)来实现高的泛化精度。图4-4展示了一个典型的两层层叠泛化方法的概念。对于第一层，基于原始数据运用不同学习参数 $\alpha=\{\alpha_1,\cdots,\alpha_L\}$ 来设计多分类器 $H=\{h_1,\cdots,h_L\}$。实际上，学习参数 α 可以用不同方式得到，如不同数据空间采样、子空间采样和基本算法框架参数等。接着，这些分类器的输出和相应的正确分类别标签被传递到第二层分类器 h'(meta-分类器)。因此，层叠泛化的主要思想是用中间分类器在不同层次学习。在具体实现细节上，普遍采用交叉验证技术在不同层次选择数据样本(Wolpert，1992)。

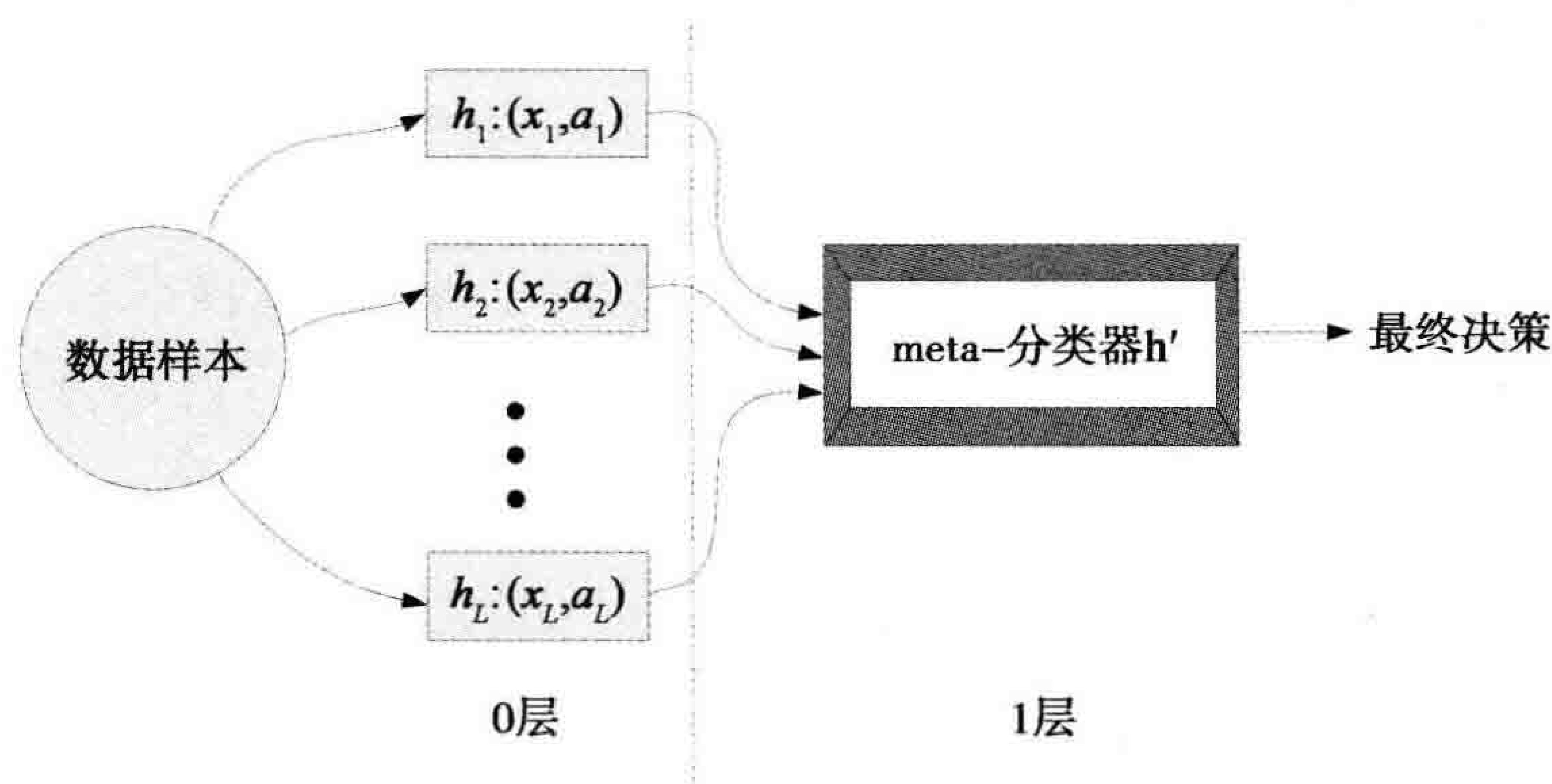


图4-4 层叠泛化方法(meta学习方法)

4.3.5 专家混合体

类似于层叠泛化方法，专家混合体(Jacobs，Jordan，Nowlan & Hinton，1991；Jordan & Jacobs，1991；Jordon & Xu，1995)也设计了两层的学习阶段，如图4-5所示。专家混合体的第一层分类器 $H=\{h_1,\cdots,h_L\}$ 的设计与层叠泛化的方法相似，但关键思想是使用一个门限网络所提供的权重分布来组合第一层分类器的输出。为此，专家混合体中的门限网络的输入来自原始数据样本，而不是像层叠泛化方法，来自第一层分类器的输出。因此，专家混合体也可以被看作是一个分类器选择算法，其组合/选择行为像一个多输入单输出的随机开关，以选择最合适的分类器或通过权重分布整合所有的分类器。特别地，门限网络是通过期望最大化(EM)技术训练得到的(Jacobs等，1991；Jordan & Jacobs，1991；Jordan & Xu，1995)。

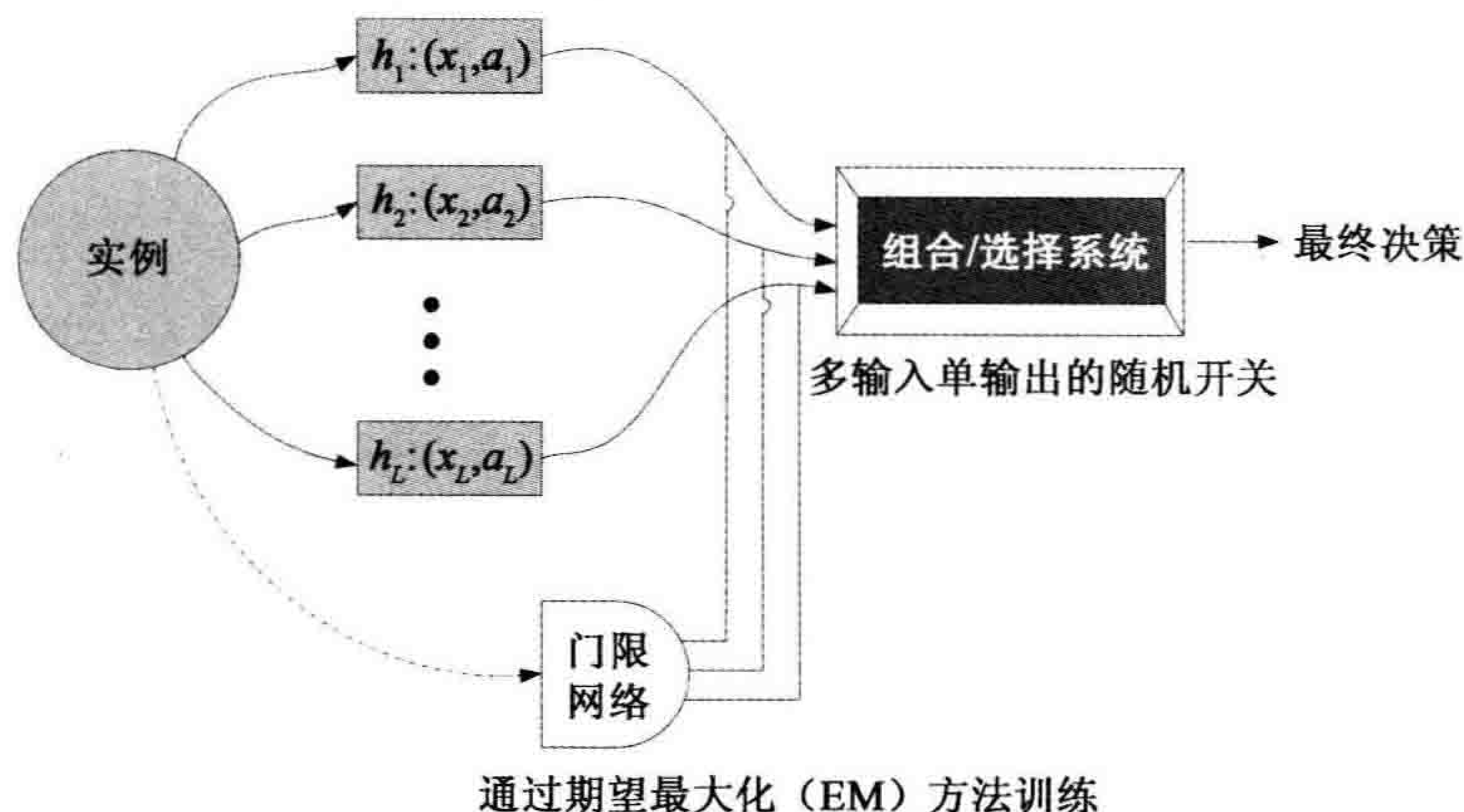


图 4-5 专家混合体

4.4 多分类器集成

集成学习的第二个重要问题是：整合多分类器的输出，以获得最终决策。假定一个集成系统具有 L 个分类器 $H=\{h_1,\cdots,h_L\}$。对于每个测试样本 x_t，每个分类器可以根据所有可能的类别标签 y_j 的后验概率 $P_i(y_i|x_t), i=1,\cdots,L$ 和 $y_j\in\Omega=\{1,\cdots,C\}$投票。这里的目标是基于每个分类器 h_i 的后验概率 $P_i(y_i|x_t)$寻求集成策略，以得到一个有改善的最终后验概率估计 $P(y_i|x_t)$。关于各种投票方法的详细讨论见 Theodoridis 和 Koutroumbas(2006)与 Kittler(1998)。

1. 几何平均规则(GA)

GA 规则从所有后验概率中寻找 $P(y_i|x_t)$，使得平均 Kullback-Leibler(KL)距离最小：

$$D_{\text{av}}=\frac{1}{L}\sum_{i=1}^{L}D_i \tag{4-23}$$

其中，

$$D_i=\sum_{y_j=1}^{C}P(y_j\mid x_t)\ln\frac{P(y_j\mid x_t)}{P_i(y_j\mid x_t)} \tag{4-24}$$

引入拉格朗日乘数并考虑 $\sum\limits_{y_j=1}^{C}P(y_j|x_t)=1$,对 $P(y_j|x_t)$ 进行优化，则式(4-23)可以表示为

$$P(y_j\mid x_t)=\frac{1}{A}\prod_{i=1}^{L}(P_i(y_j\mid x_t))^{1/L} \tag{4-25}$$

其中，A 为独立类的数量。

基于式(4-25)，GA 规则预测测试样本 x_t 对应的类别标签 y_j，以使 $P_i(y_j\mid x_t)$

的积最大。

GA 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j} \prod_{i=1}^{L} P_i(y_j \mid x_t) \tag{4-26}$$

2. 算术平均规则(AA)

不同于式(4-24)，也可以定义如下概率距离来替代 KL 距离：

$$D_i = \sum_{y_j=1}^{C} P_i(y_j \mid x_t) \ln \frac{P_i(y_j \mid x_t)}{P(y_j \mid x_t)} \tag{4-27}$$

将式(4-27)带入式(4-23)中，可以得到：

$$P(y_j \mid x_t) = \frac{1}{L} \sum_{i=1}^{L} P_i(y_j \mid x_t) \tag{4-28}$$

因此，AA 规则可以被定义为寻找 $P_i(y_j \mid x_t)$ 算术平均值的最大值。

AA 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j} \frac{1}{L} \sum_{i=1}^{L} P_i(y_j \mid x_t) \tag{4-29}$$

3. 中值规则(MV)

当概率 $P_i(y_j \mid x_t)$ 为异常值时，由于异常值会主导投票过程，因此 AA 规则会导致组合性能低下。在这种情况下，MV 规则将预测具有最大中值的最终类别标签。

MV 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j} \{\text{median}(P_i(y_j \mid x_t))\} \tag{4-30}$$

4. 多数投票规则(MajV)

不像软型规则(soft-type rule)，如 GA 和 AA，MajV 规则是一个硬集成策略。基于训练信息，假定每个分类器对测试样本 x_t 预测一个类别标签，MajV 规则简单地输出在所有类中收到选票最多的标签作为最终预测标签。如果有多个相同的类别标签的票数且最大，那么可在其中随机选择一个作为最终的类别标签。MajV 规则定义如下。

MajV 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j} \frac{1}{L} \sum_{i=1}^{L} \Delta_i(y_j \mid x_t) \tag{4-31}$$

其中，

$$\Delta_i(y_j \mid x_t) = \begin{cases} 1, & h_i(x_t) = y_j \\ 0, & \text{其他} \end{cases}$$

5. 最大值规则

最大值规则的基础是所有可能的类别标签的 $P_i(y_j \mid x_t)$ 的最大值所提供的信息。

与基于 $P_i(y_j|x_t)$均值的 AA 规则不同，最大值规则更像是一个胜者为王的投票方式：

最大值规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j}\{\max(P_i(y_j|x_t))\} \tag{4-32}$$

6. 最小值规则

与最大值规则相似，最小值规则是在所有类别标签所对应的 $P_i(y_j|x_t)$的最小值中找出最大值。类似于式(4-32)，最小值规则的定义如下。

最小值规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j}\{\min(P_i(y_j|x_t))\} \tag{4-33}$$

7. 波达计数法(BC)规则

BC 规则的基础是由单个 $P_i(y_j|x_t)$所提供的类别标签的排序顺序。根据分类器的输出，每个分类器对所有可能的类别标签排序。对一个 C 类问题，排名为 k 的候选者从最终投票系统中接收$(C-k)$个选票。最后，收到最多选票的类别标签将是最终的预测结果。BC 规则的定义如下。

BC 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j}\sum_{i=1}^{L}\Omega_i(y_j|x_t) \tag{4-34}$$

其中，对于类别标签 y_j，如果分类器 h_i 把 x_t 排在第 k 位，那么 $\Omega_i(y_j|x_t)=C-k$。

8. 加权规则

对于最终投票结果，为了反映不同分类器的不同置信水平，在上述方法中可以对每个分类器独立引入权重系数。这里定义了两种常用的方法。

加权 AA 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j}\frac{1}{L}\sum_{i=1}^{L}\omega_i \cdot P_i(y_j|x_t) \tag{4-35}$$

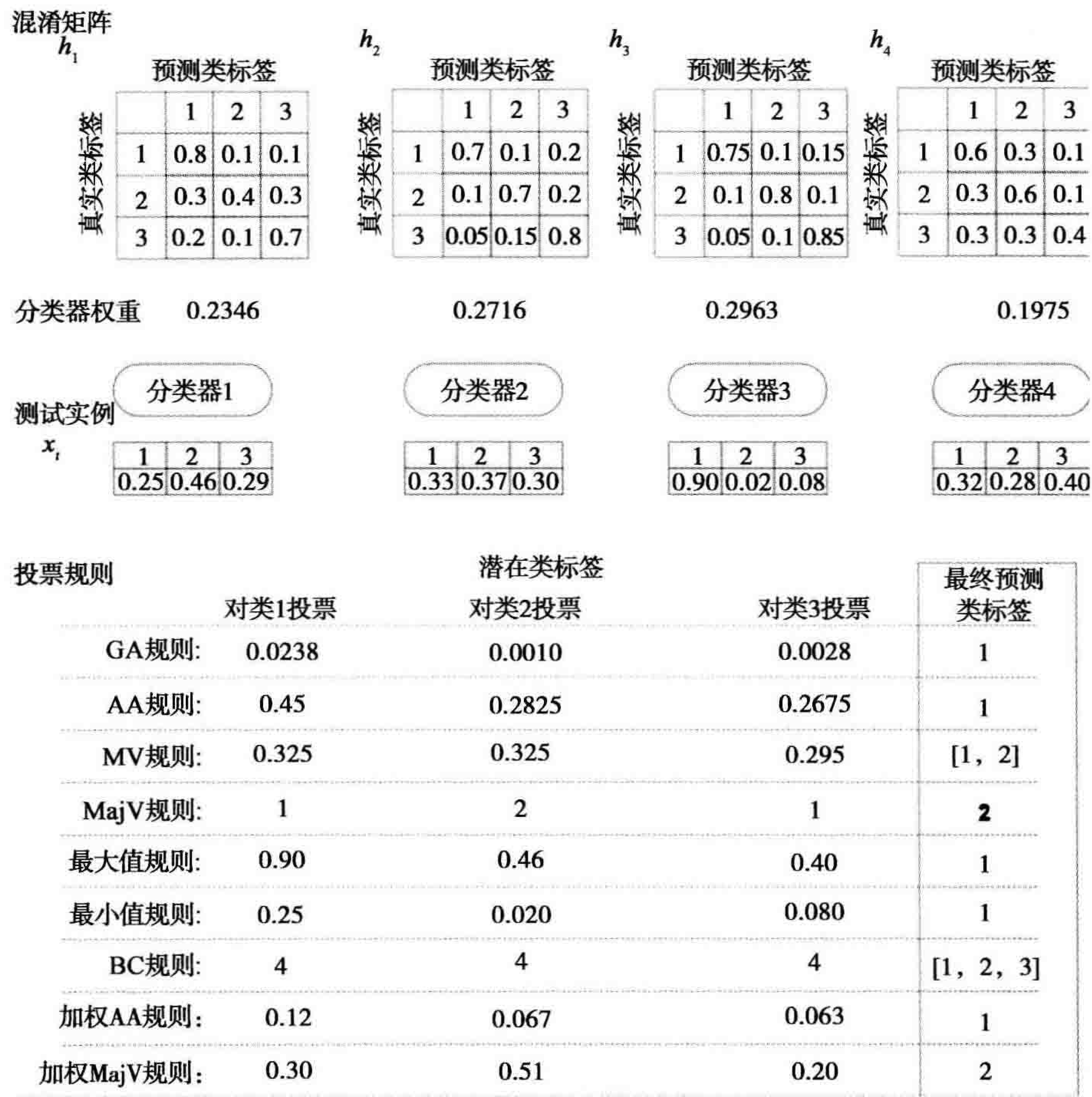
加权 MajV 规则：

$$x_t \rightarrow y_j \text{ 满足} \max_{y_j}\sum_{i=1}^{L}\omega_i \cdot \Delta_i(y_j|x_t) \tag{4-36}$$

其中，ω_i 是分类器 h_i 的权重系数：$\omega_i \geqslant 0$ 且 $\sum_{i=1}^{L}\omega_i = 1$。确定权重系数的常用方法是交叉验证技术(Theodoridis & Koutroumbas，2006)。

为了综述这些方法，图 4-6 给出了一个三类分类问题($C=3$)的例子。在这里，假设分类器集成学习系统包括 4 个分类器：h_1、h_2、h_3 和 h_4。对于加权方法(加权 AA 规则和加权 MajV 规则)，权重系数是由分类器在训练数据集上产生的混淆矩阵

决定的。从图 4-6 可以看出，在这个特定样本中，MajV 规则和加权 MajV 规则将测试样本 x_t 投票为第 2 类标签。对于 MV 规则，由于类别 1 和类别 2 的投票是相同的，最终预测标签可以从两者中随机选择。对于 BC 规则，最终预测标签可以从类 1、类 2 和 3 类中随机选择。其他的方法将测试样本投票为第 1 类标签。



投票规则	潜在类标签 对类1投票	对类2投票	对类3投票	最终预测类标签
GA规则:	0.0238	0.0010	0.0028	1
AA规则:	0.45	0.2825	0.2675	1
MV规则:	0.325	0.325	0.295	[1，2]
MajV规则:	1	2	1	2
最大值规则:	0.90	0.46	0.40	1
最小值规则:	0.25	0.020	0.080	1
BC规则:	4	4	4	[1，2，3]
加权AA规则:	0.12	0.067	0.063	1
加权MajV规则:	0.30	0.51	0.20	2

图 4-6 集成多分类器的技术概括

4.5 实例研究

为了说明集成学习方法在现实世界中的应用，本节给出本章所讨论的各种集成学习方法的实例研究。

4.5.1 数据集和实验配置

当前的标准测试使用了 UCI 机器学习库(Asuncion & Newman，2007) 的 20 个

基准数据集。表 4-2 说明了该数据集的特征。

基础学习算法为多层感知器神经网络模型，其中输入神经元、输出神经元的数量分别等于每个数据集的属性、类别的数量。表 4-2 给出了对应每个数据集的隐层神经元数目，满足条件 $N=O\left(\frac{W}{\varepsilon}\right)$(Haykin，1999)，其中 N 为训练集的大小，$O(\cdot)$表示阶数，W 为模型中自由参数的总数量，ε 表示期望的测试误差率，本例设为 0.1。用 Sigmoid 函数作为激活函数，用反向传播训练神经网络。本例展示的所有结果取自随机运行 100 次的结果平均值。每次运行时，在数据集中随机选择一半(50%)的数据作为训练集，另一半(50%)的数据作为测试集。用 Bagging 方法建立分类器集成系统(Breiman，1996)。对于集成学习方法，Opitz 和 Maclin(1999)提出使用 25 个集成分类器可获得相对大的收益，因此在本样本研究中，在每次运行中可通过自举采样构建 25 个分类器。

表 4-2 在样本研究中使用的数据集的特征

数据集			属性		隐层
名称	样本个数	类别数	连续	离散	神经元数
ecoli	336	8	7	0	2
german	1000	2	7	13	2
glass	214	7	9	0	2
haberman	306	2	0	3	10
ionosphere	351	2	34	0	2
iris	150	3	4	0	10
letter-recognit	20 000	26	16	0	10
muskl	476	2	166	0	2
pima-indians-di	786	2	8	0	10
satimage	6435	6	36	0	10
segmentation	2310	7	19	0	10
shuttle	59 000	2	9	0	2
sonar	208	2	60	0	2
soybean-small	47	4	0	35	2
spectf	267	2	44	0	2
vehicle	846	4	18	0	2
vowel	990	11	10	0	2
wdbc	569	2	30	0	2
wine	178	3	13	0	2
yeast	1484	10	8	0	10

4.5.2 仿真结果

表4-3给出了4.4节中所讨论的多数组合投票方法的测试误差率性能。对于每个数据集，带有下划线的为获胜投票策略。此外，每种投票策略在所有数据集上获胜的总次数见表4-3的底部。一个显而易见的问题是："对于这些集成组合方法，是否有最好的方法(或哪一个是最好的方法)?"在机器学习领域，关于这个问题存在许多有趣的讨论(Kittler等，1998；Polikar，2006；Dietterich，2000a，2000b；Breiman，1998；Bauer & Kohavi，1999；Quinlan，1996；Drucker，Cortes，Jackel，LeCun & Vapnik，1994；Battiti & Colla，1994；Kuncheva，2002；Tax，van Breukelen，Duin & Kittler，2000；Jacobs，1995；Duin & Tax，2000)。总之，"没有免费的午餐"定理(Wolpert & Macready，1997)证明：事实上没有最佳方法。相反，每个问题的最好解决方法可能依赖于特定领域的知识和数据特征。为了更深入地讨论这个问题，接下来进行间隔分析。

表4-3 测试误差率性能

投票方法	GA规则	AA规则	MV规则	MajV规则	Max规则	Min规则	BC规则	加权AA规则	加权MajV规则
ecoli	19.82	21.07	21.26	21.39	22.94	**17.53**	19.79	21.07	21.39
german	**24.35**	24.44	24.48	24.48	24.47	24.47	24.48	24.43	24.48
glass	**43.37**	43.87	44.88	45.06	44.36	43.48	44.01	43.53	44.66
haberman	25.50	25.61	25.82	25.82	**25.44**	**25.44**	25.82	25.65	25.82
ionosphere	13.32	12.99	12.88	12.88	**11.34**	**11.34**	12.88	12.95	12.88
iris	**3.55**	**3.55**	3.63	3.63	3.59	3.57	3.63	3.57	3.63
letter-recognit	**38.36**	40.10	42.56	51.26	49.32	39.64	43.00	39.61	48.47
muskl	**23.90**	24.17	25.24	25.24	25.82	25.82	25.24	23.97	25.10
pima-indians-di	32.59	32.65	33.14	33.14	**32.23**	**32.23**	33.14	32.65	33.14
satimage	4.87	6.33	9.09	8.27	6.51	**2.88**	4.04	5.05	6.16
segmentation	**12.29**	13.06	14.24	20.05	16.15	12.32	14.71	12.92	17.65
shuttle	7.23	7.22	7.99	7.89	6.22	**4.07**	8.10	7.22	7.80
sonar	22.23	21.63	**21.51**	**21.51**	25.09	25.09	**21.51**	21.61	**21.51**
soybean-small	1.54	1.42	1.29	1.17	7.25	3.71	1.46	1.33	**1.13**
spectf	21.00	**20.98**	**20.98**	**20.98**	22.48	22.48	**20.98**	**20.98**	**20.98**
vehicle	35.23	36.02	46.11	49.23	38.69	**35.09**	46.55	36.13	49.22
vowel	44.34	45.21	46.11	46.37	46.74	**44.28**	44.96	45.31	46.32
wdbc	13.61	20.42	37.20	37.20	**8.57**	**8.57**	37.20	13.92	34.86
wine	29.47	34.26	47.88	50.04	28.56	**17.66**	48.04	30.89	45.42
yeast	40.02	40.00	40.08	40.09	40.49	40.47	40.05	**40.00**	40.08
Winning times	**6**	**2**	**2**	**2**	**4**	**10**	**2**	**2**	**3**

4.5.3 间隔分析

间隔分析(margin analysis)(Schapire, Freund, Bartlett & Lee, 1998) 将集成学习和最先进的统计学习方法连接起来，并提供了许多重要的深入见解，例如 SVM (Vapnik, 1995, 1998)。本节用一些理论分析和实例分析来研究集成学习中的间隔分析。

1. 间隔分析简史

一般来说，在学习系统(Schapire 等，1998)中，大间隔对投票噪声扰动具有高容忍性。通过最大化间隔来提高分类器的泛化能力的思想，可以追溯到 Vapnik (1995) 的经典著作《统计学习理论》，该著作为最优间隔分类器(Boser, Guyon, & Vapnik, 1992)和 SVM(Cortes & Vapnik, 1995)奠定了基础。这些方法的主要思想之一是将低维空间中的非线性分类器转化成高维空间中的线性分类器，通常用核方法实现。

在间隔分析方面，Schapire 等(1998)建立了自举方法(Freund & Schapire, 1996, 1997)与最大间隔分类器的重要联系。这两种方法的目标都是寻找高维空间中最大间隔的线性组合。当然，应该注意到这两种机制的差异：SVM 用核方法实现高维空间的有效计算，而自举方法用一个基学习分类器建立某一时刻的高维空间坐标(Schapire 等，1998)。此外，这些方法的优化目标也是不同的：SVM 的目标是基于支撑向量使最小间隔最大化，而自举方法的目标是使样本的指数权重最小化(Schapire 等，1998)。

现有的许多研究讨论了基于间隔的分类器并分析了间隔特征。例如，前文提到的在 Schapire 等人(1998)对投票方法的间隔展开讨论之后不久，Breiman(1999)严重质疑他们对间隔的解释，并提出了一个新的泛化误差率上界。他们在论文中提出的一种称为 arc-gv 的自举型算法证明了他们的泛化误差率界比 Schapire 等给出的更加严格。但是，尽管 arc-gv 方法能够产生比 AdaBoost 方法更高的间隔分布，但其性能却不如 AdaBoost。因此，Breiman 质疑 Schapire 等人的间隔分析，并指出了 VC 型界限具有误导作用。最近，Reyzin 和 Schapire(2006) 重新考虑这一问题，经过仔细研究 arc-gv 方法，获得了很多深入的发现。Reyzin 和 Schapire 的主要发现是分类器的复杂性分析，他们在解释 arc-gv 方法的性能较差的原因是：由于增加了基分类器的复杂性。Reyzin 和 Schapire 指出，这不仅解释了 Breiman 的疑惑，而且也符合 Schapire 等人提出的间隔理论。因此，最大化间隔仍然是可用的，但在以基分

类器的复杂性为代价(Reyzin & Schapire，2006)时没有必要使用。最近，Wang，Sugiyama，Yang，Zhou 和 Feng(2008)对这个问题给出了另一种有趣的讨论，他们的论文提出了一种新间隔度量界限，称为均衡间隔(emargin)，并基于 AdaBoost 方法和 arc-gv 方法进行了详细分析。

在自举算法分析与解释间隔方面有很多研究工作。例如，Rätsch 和 Warmuth(2001) 提出了 AdaBoost 学习方案的一个变体——间隔自举算法。该算法被证明能够快速收敛于最大间隔。Warmuth、Glocer 和 Rätsch(2008) 基于 LPBoost 算法提出了另一种基于间隔的集成学习方法——SoftBoost 方法。该方法在实际的自举算法中实现了软间隔的思想，并致力于优化软间隔。Rätsch、Mika、Schölkopf 和 Müller(2002) 宣称，一个 SVM 算法可以转换为类自举(boosting-like)算法，反之亦然。基于这种关系提出了一种类自举单类杠杆算法。此外，Lin 和 Li(2008)提出了一种基于 SVM 的无限集成学习框架，同时也阐述了集成学习和 SVM 之间的关系。Romero、Carreras 和 Marquez(2004) 针对真实世界的分类应用，比较了 3 种基于间隔的学习方法——SVM、AdaBoost 和前馈神经网络的实验结果(即文本分类问题)。其他有趣的研究包括数据依赖间隔泛化界限分类(Antos，Kégl，Linder & Lugosi，2002)、复杂分类器泛化误差率的上置信界限的复杂性度量和分析(Koltchinskii & Panchanko，2002，2005)和基于阈值凸组合分析的新泛化误差率界限(Mason，Bartlett & Golea，2002)等。

2. 集成学习的间隔分析

考虑一个两类分类问题。假定训练数据 D_{tr} 中所有样本都可以被表示为$\{x_j, y_j\}$，$j=1,\cdots,m$，并且 $y_j\in\{-1,+1\}$，是 x_j 的类别标签。进一步假定，$h(x)$是某种实现样本到高维空间的固定非线性映射。因此，通过向量 σ 可以定义最大间隔分类器，从而使式(4-37)最大化(Schapire 等，1998)。

$$\min_{(x,y)\in D_{tr}} \frac{y(\sigma\cdot h(x))}{\|\sigma\|_2} \tag{4-37}$$

其中，$\|\sigma\|_2$ 是向量 σ 的 l_2 范数。因此，该方法的目标是在高维空间中寻找最优超平面，以使最小间隔最大化。另一方面，AdaBoost 方法的关键思想是在训练数据集 D_{tr}上迭代更新分布函数。这样，对于每一次迭代 $t(=1,\cdots,L$，其中 L 是预先设置的总迭代次数)，分布函数 D_{tr}被持续更新并被用来训练一个新分类器：

$$D_{t+1}(j) = \frac{D_t(j)\exp(-y_j\sigma_t h_t(x_j))}{Z_t} \tag{4-38}$$

其中，$\sigma_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，$h_t(x_j)$是分类器 h_t 对样本 x_j 的预测输出，ε_t 是分类器 h_t

在训练集上的误差率：$\varepsilon_t = \sum_{j:h_t(x_j)\neq y_j} D_t(j)$，并且 $Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$ 是归一化因子，D_{t+1} 为分布函数，即 $\sum_{j=1}^{m} D_{t+1}(j) = 1$（更多细节见 4.3.2 节）。这样，最终的组合分类器是通过一系列已有的分类器加权投票得到的(Schapire 等，1998)。

$$f(x) = \frac{\sum_{t=1}^{L}\sigma_t h_t(x)}{\sum_{t=1}^{L}\sigma_t} \tag{4-39}$$

Schapire 等(1998)指出，如果将系数 $\{\sigma_t\}_{t=1}^{L}$ 作为向量 $\sigma \in \mathscr{R}^L$ 的坐标，将分类器的输出 $\{h_t(x)\}_{t=1}^{L}$ 作为向量 $h(x) \in \{-1, +1\}^L$ 的坐标，则等式(4-39)可以被重写为

$$f(x) = \frac{\sigma h(x)}{\|\sigma\|_1} \tag{4-40}$$

其中，$\|\sigma\|_1$ 为 σ $\left(\|\sigma\|_1 = \sum_{t=1}^{L}|\sigma_t|\right)$ 的 l_1 范数。

比较式(4-37)和式(4-40)，可以清楚地看出自举方法与最大间隔分类器之间的联系(Schapire 等，1998)：两种方法的目标都是在高维空间中寻找较大间隔的线性组合。当然，正如在“间隔分析简史”小节提及的那样，其具体实现和计算细节还是有所不同的。例如，SVM 的目标是基于支撑向量使最小间隔最大化，而自举方法的目标是使样本的指数权重最小化。关于学习机制，SVM 依赖核函数技术在高维空间中实现有效计算，而自举方法依赖于一个基学习分类器，以建立某一时刻的高维空间坐标。

本节采用间隔和间隔分布曲线(Schapire 等，1998)对集成学习中的间隔分析进行形式化讨论。

定义 1：考虑一个分类问题，样本的分类间隔定义为：分配给正确标签的权重与分配给每个错误标签的最大权重之间的差异，也就是，对于样本(x, y)，有

$$\text{margin}(x) = w_{h(x)=y} - \max\{w_{h(x)\neq y}\} \tag{4-41}$$

定义 2：给定一个数据分布 D，间隔分布曲线被定义为

$$F(\lambda) = \frac{|D_\lambda|}{|D|}, \lambda \in [-1,1] \tag{4-42}$$

其中，$|D_\lambda = \{x: \text{margin}(x) \leqslant \lambda\}|$，$|\cdot|$ 为计算样本数量的操作，$F(\lambda) \in [0,1]$。

基于这些定义，图 4-7 给出了 AA 规则、MV 规则和 AdaBoost. M1 方法的间隔分布曲线的例子，测试数据取自 UCI 机器学习知识库(Asuncion & Newman，2007)的“葡萄酒”数据集。其中 x 轴为式(4-41)中定义的间隔，y 轴为基于式(4-42)

的累积分布。从该图可以看出，AdaBoost. M1 方法取得了高间隔，间隔小于 0.6 的测试数据占 65.20%，如图 4-7 中的实线所示，而对于 MV 规则和 AA 规则，间隔小于 0.6 的测试数据分别为 95.18% 和 97.74%。注意，在间隔分布曲线中，对应于 0 间隔值的累积分布值(y 轴)(见图 4-7 方框所示)表示分类误差率。对于 AdaBoost. M1 方法、AA 规则和 MV 规则，各自对应的测试数据的分类误差率分别为 29.25%、34.26%和 48.49%。

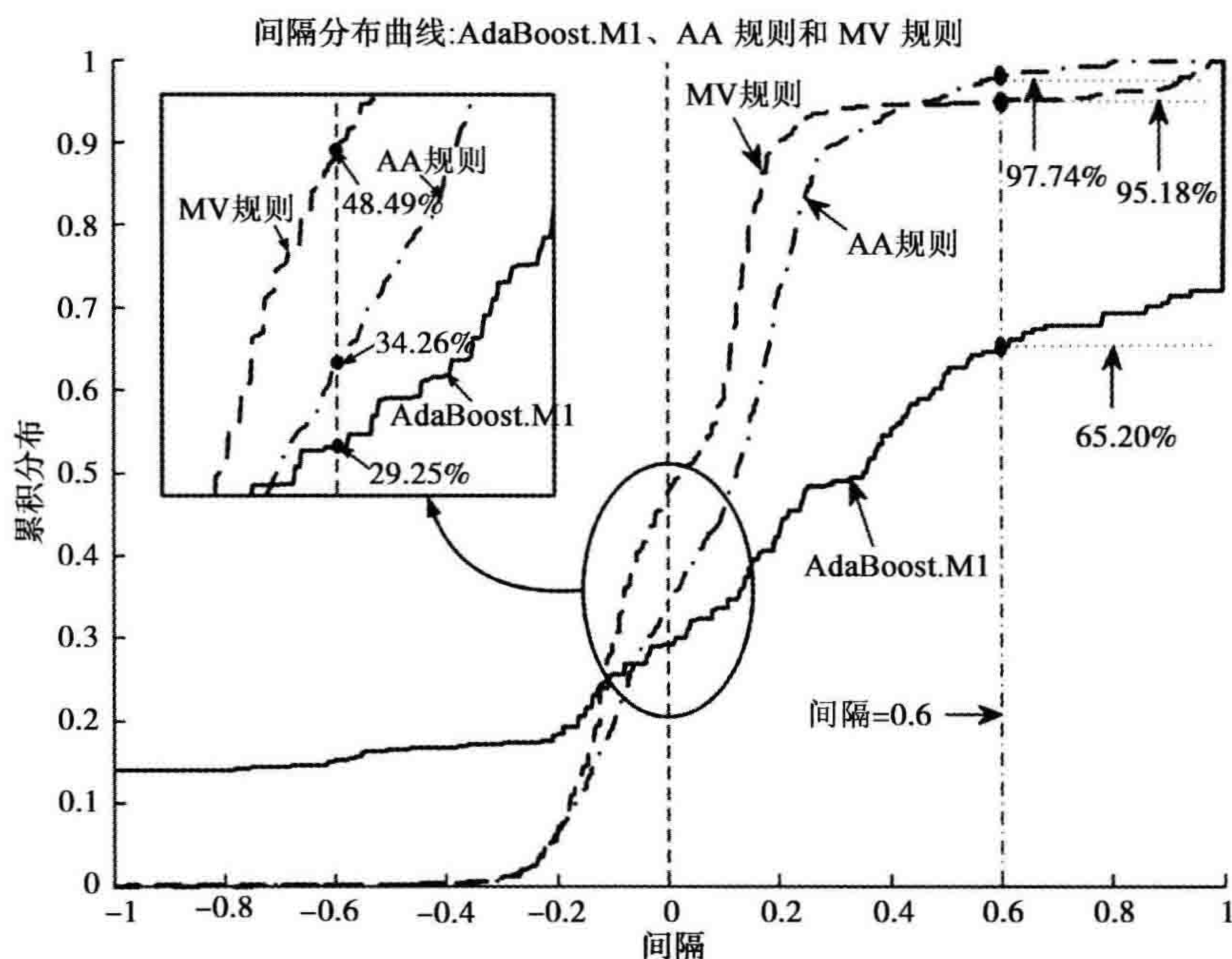


图 4-7　AdaBoost. M1 方法、AA 规则和 MV 规则的间隔分析

为了对主要的集成学习方法进行详细的间隔分析，我们分析了分类器集成方法和自举方法(AdaBoost. M1 方法和 AdaBoost. M2 方法)的间隔分布曲线。图 4-8a～d 展示了在“玻璃”和“车辆”数据集上的曲线分析结果。从图中可以看出，随着数据样本间隔的增加，AdaBoost. M1 方法表现得较为激进：间隔分布曲线显著地变化到 1 和−1。仿真结果和观察都与 Schapire 等(1998)关于自举方法特性的分析是一致的。正如 Schapire 等人指出，大间隔对投票噪声扰动具有高容忍性。希望本节的分析，能为读者集成学习中的间隔分析提供一些有用的帮助。

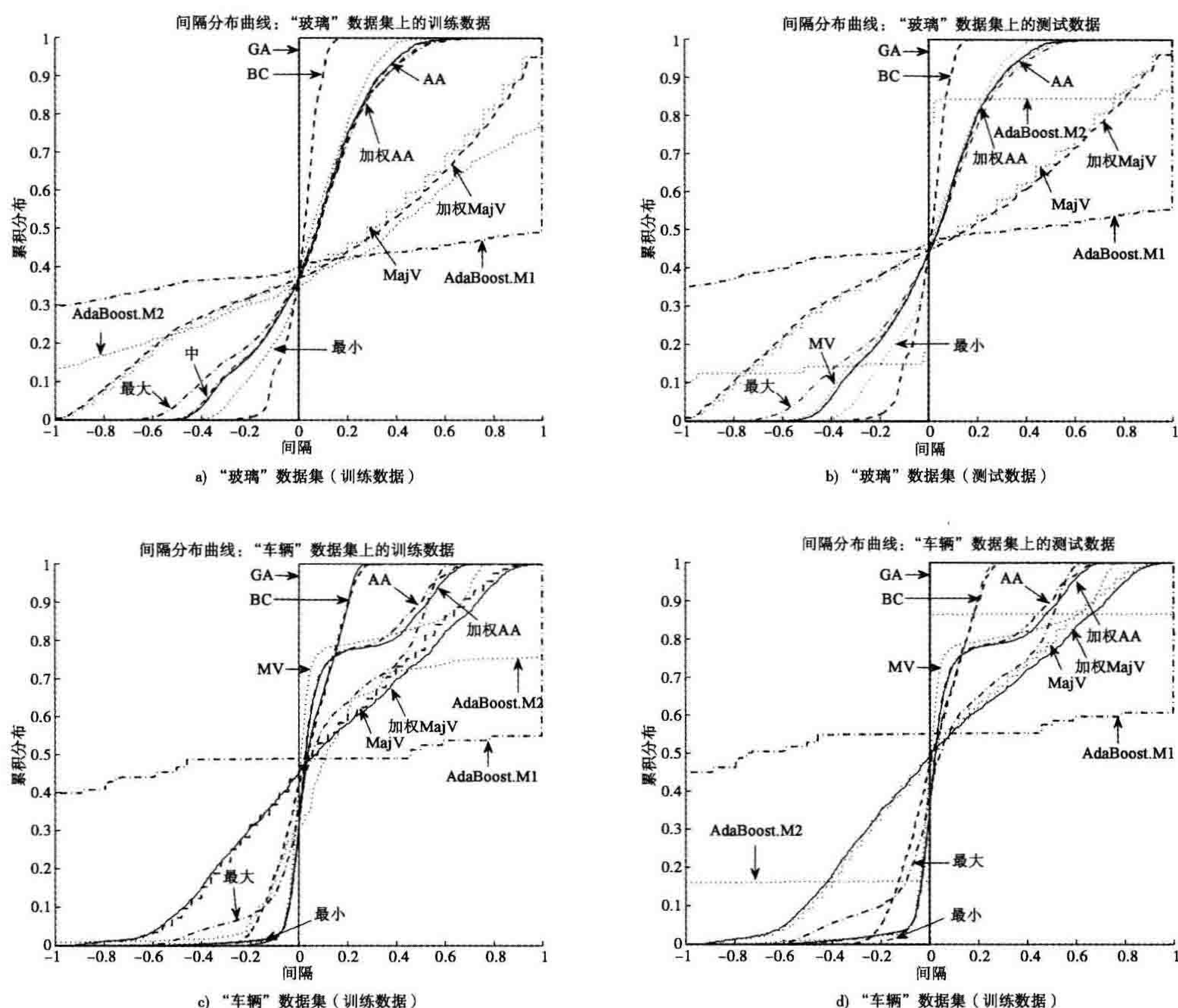


图 4-8 间隔分析

4.6 总结

本章讨论了机器智能研究中的集成学习，要点包括：

- 一个集成学习系统通常包括两个主要过程：设计多个多样化的分类器，集成多个分类器来支持最终决策过程。
- 分类器的多样性对于一个成功的集成学习方法至关重要。有许多度量可以用于评估分类器的多样性，包括 Q 统计量(Q)、相关性系数(ρ)、不一致度量($\xi_{i,j}$)、双错度量(γ)、熵协议(E)、Kohavi-Wolpert 方差(KW)、测试者间的一致性(k)、困难程度(θ)、广义多样性和一致失败多样性(GD/CFD)。所有这些度量可定量评估分类器的多样性。
- 多样性的多分类器有多种设计方法，代表性的研究工作包括自举聚集、自适应自举、子空间学习(即决策森林、随机森林、排序子空间)、层叠泛化和专家混合体。

- 如何集成多分类器的输出以支持最终决策是集成学习的另一个重要问题。常用的组合投票方法包括 GA 规则、AA 规则、MV 规则、MajV 规则、Max 规则、Min 规则、BC 规则和各种加权规则。一般来说，每个领域都不存在单一的最好组合规则，每个问题的最好解决方法应该依赖于特定领域的知识和数据特征。
- 间隔分析能够为集成学习方法提供重要的洞察和有价值的建议。一般来说，大间隔对投票噪声扰动具有高容忍性。通过最大化间隔来提高分类器的泛化性的思想可以追溯到《统计学习理论》经典著作，该著作为最优间隔分类器和 *SVM* 奠定了基础。本章为集成学习的间隔分析提供了理论分析和实例研究。

参考文献

Antos, A., Kégl, B., Linder, T., & Lugosi, G. (2002). Data-dependent margin-based generalization bounds for classification. *Journal of Machine Learning Research*, *3*, 73–98.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository [Online], Available: http://www.ics.uci.edu/~mlearn/MLRepository.html*.

Battiti, R., & Colla, A. M. (1994). Democracy in neural nets: Voting schemes for classification. *Neural Networks*, *7*(4), 691–707.

Bauer, E., & Kohavi, R. (1999). An empiricial comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, *36*, 105–142.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proc. Annual ACM Workshop on Computational Learning Theory*, pp. 144–152.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (1998). Arcing classifiers. *Annals of Statistics*, *26*(3), 801–849.

Breiman, L. (1999). Prediction games and arcing algorithms. *Neural Computation*, *11*, 1493–1517.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, *20*, 273–297.

Cunningham, P., & Carney, J. (2000). Diversity versus quality in classification ensembles based on feature selection. *Technical Report TCD-CS-2000-02*. Department of Computer Science, Trinity College Dublin.

Dietterich, T. (2000a). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Information & Software Technology*, *40*(2), 139–157.

Dietterich, T. G. (2000b). Ensemble Methods in Machine Learning. In J. Kittler and F. Roli (Ed.) *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Sceince*, vol. 1857 (pp. 1–15). New York: Springer Verlag.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, *6*(6), 1289–1301.

Duin, R., & Tax, D. (2000). Int. workshop on multiple classifier systems, lecture notes in computer science. In J. Kittler & F. Roli (Eds.), (Vol. 1857, pp. 16–29). Springer.

Fleiss, J. (1981). *Statistical methods for rates and proportions*. New York: Wiley.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proc. Int. Conf. Machine Learning*, pp. 148–156.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Giacinto, G., & Roli, F. (2001). Design of effective neural network ensembles for image classification processes. *Journal of Image Vision and Computing*, *9*, 699–707.

Hansen, L., & Salamon, P. (1990). Neural network ensembles. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001.

Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

He, H., & Shen, X. (2007). A ranked subspace learning method for gene expression data classification. *Proc. Int. Conf. Artificial Intelligence*, pp. 358–364.

Ho, T. K. (1998a). C4.5 decision forests. *Proc. Int. Conf. Pattern Recognition*, pp. 545–549.

Ho, T. K. (1998b). Random subspace method for constructing decision forests. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Ho, T. K. (1995). Random decision forests. *Proc. Int. Conf. Document Analysis and Recognition*, pp. 278–282.

Jacobs, R. A. (1995). Methods for combining experts' probability assessments. *Neural Computation*, *7*(5), 867.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*(1), 79–87.

Jordan, M. J., & Jacobs, R. A. (1991). Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, *6*(2), 79–87.

Jordan, M. J., & Xu, L. (1995). Convergence results for the em approach to mixtures of experts architectures. *Neural Networks*, *8*(9), 1409–1431.

Kittler, J., Hatel, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *20*(3), 226–239.

Kohavi, R., & Wolpert, D. (1996). Machine learning: Proc. 13th international conference. In L. Saitta (ED.), (pp. 275–283). Morgan Kaufmann.

Koltchinskii, V., & Panchanko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, *30*, 1–50.

Koltchinskii, V., & Panchanko, D. (2005). Complexities of convex combinations and bounding the generalization error in classification. *Annals of Statistics*, *33*, 1455–1496.

Krogh, A., & Vedelsby, J. (1995). Advances in neural information processing systems. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), (Vol. 7, pp. 231–238). Cambridge, MA: MIT Press.

Kuncheva, L. I. (2002). Switching between selection and fusion in combining classifiers: An experiment. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, *32*(2), 146–156.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*, 181–207.

Kuncheva, L. I., Whitaker, C., Shipp, C., & Duin, R. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, *6*, 22–31.

Lam, L. (2000). Int. workshop on multiple classifier systems, lecture notes in computer science. In J. Kittler & F. Roli (Eds.), (Vol. 1857, pp. 78–86). Springer.

Lin, H.-T., & Li, L. (2008). Support vector machinery for infinite ensemble learning. *Journal of Machine Learning Research*, *9*, 285–312.

Littlewood, B., & Miller, D. (1989). Conceptual modeling of coincident failures in multiversion software. *IEEE Trans. Software Engineering*, *15*(12), 1596–1614.

Looney, S. (1988). A statistical technique for comparing the accuracies of several classifiers. *Pattern Recognition Letters*, *8*, 5–9.

Mason, L., Bartlett, P., & Golea, M. (2002). Generalization error of combined classifiers. *Journal of Computer and System Sciences*, *65*, 415–438.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *J. Artificial Intelligence Research*, *11*, 169–198.

Partridge, D., & Krzanowski, W. J. (1997). Software diversity: Practical statistics for its measurement and exploitation. *Information & Software Technology*, *39*, 707–717.

Polikar, R. (2006). Esemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45.

Quinlan, J. R. (1996). Bagging, boosting and c4.5. *Proc. Int. Conf. on Artificial Intelligence*, pp. 725–730.

Rätsch, G., Mika, S., Schölkopf, B., & Müller, K.-R. (2002). Constructing boosting algorithms from svms: an application to one-class classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *24*(9), 1184–1199.

Rätsch, G., & Warmuth, M. K. (2001). Marginal boosting. *NeuroCOLT2 Technical Report Series NC2-TR-2001-097*.

Reyzin, L., & Schapire, R. E. (2006). How boosting the margin can also boost classifier complexity. *Proc. Int. Conf. Machine Learning*, 148, 753–760.

Romero, E., Carreras, X., & Marquez, L. (2004). Exploiting diversity of margin-based classifiers. *IEEE International Joint Conference on Neural Networks*, 1, 419–424.

Rosen, B. (1996). Ensemble learning using decorrelated neural networks. *Connection Science*, *8*(3/4), 373–383.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, *26*(5), 1624–1686.

Skalak, D. (1996). The sources of increased accuracy for two proposed boosting algorithms. *Proc. Association for the Advancement of Artificial Intelligence Conf*.

Sneath, P., & Sokal, R. (1973). *Numerical taxonomy*. New York: W. H. Freeman..

Tax, D. M. J., van Breukelen, M. Duin, R. P. W., & Kittler, J. (2000). Combining multiple classifiers by averaging or by multiplying? *Pattern Rec.*, *33*(9), 1475–1485.

Theodoridis, S., & Koutroumbas, K. (2006). *Pattern recognition* (3rd ed.). New York: Elsevier.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley-Interscience.

Wang, L., Sugiyama, M., Yang, C., Zhou, Z.-H., & Feng, J. (2008). On the margin explanation of boosting algorithm. *Proc. Annual Conf. Learning Theory*, pp. 479–490.

Warmuth, M. K., Glocer, K., & Rätsch, G. (2008). Boosting algorithms for maximizing the soft margin. *Advances in Neural Information Processing Systems*.

Wolpert, D. H. (1992). Stacked generalization. *Neural Network*, *5*(2), 241–259.

Wolpert, D. H., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation*, *1*(1), 67–82.

Yang, Y., Wang, G., & He, K. (2007). An approach for selective ensemble feature selection based on rough set theory. *Lecture Notes in Computer Science*, pp. 518–525.

Yule, G. (1900). On the association of attributes in statistics. *Phil. Trans.*, *194*(A), 257–319.

第5章

机器智能的自适应动态规划

5.1 引言

正如在第1章中的讨论，脑科学研究方面强有力的证据表明生物大脑处理任务的方法完全不同于当今的计算机。从工程设计的观点来看，如果大脑被视为一个智能控制器系统(Werbos，2002，2005)，那么重要的问题是如何能够随时间动态地学习信息、积累知识、做出预测和调整行为，以使某种效用函数最大化，最终实现目标(目标导向行为)(Werbos，2004，2007，2009a)的通用方法。为了这个目标，自适应动态规划(ADP)是公认的机器智能发展的核心方法，或者说“在通常情况下，是近似最优行动策略的唯一通用学习方法”(Werbos，2004，2007，2009a)。

过去的几年里，ADP从基础理论研究到广泛应用，在机器智能领域引起了广泛的关注(Werbos，2009b；Si，barto，Powell & Wunsch，2004；Prokhorov & Wunsch，1997；White & Sofge，1992；Bertsekas & Tsitsiklis，1996；Powell，2007；Liu & Jin，2009；Wang，Zhang & Liu 2009；Vamvoudakis & Lewis，2009；Balakrishnan，Ding & Lewis，2008；Al-Tamimi，Abu-khalf & Lewis，2007；Venayagamoorthy，Harley & Wunsch，2003)。本章主要讨论ADP的学习原理、构架及其在机器智能研究方面的应用，也在现有研究结果的基础上，提出一种具有多目标表示的分层学习ADP构架，以有效地整合优化和预测，实现通用学习。

5.2 基本目标：优化和预测

在向设计通用的类脑智能系统的长期努力中，优化和预测是公认的目标导向行为的两个必不可少的重要目标(Werbos，2009a)。例如，在强化学习框架中(ADP问题的子集)，智能系统的目标是：通过与外部环境的动态交互使期望的奖励信号最

大化(Sutton & Barto，1998；Geramifard，Bowling & Sutton，2006；Rafols，Rings，Sutton & Tanner，2005；Kaelbling，Littman & Moore，1996)。因此，理解优化和预测的基本问题，将有助于研究团体聚焦于最关键和最有希望的前沿研究课题，实现基础科学突破，开发工程技术，使机器智力水平更接近实际应用需求(Werbos，2009a)。

优化在控制理论、决策理论、风险分析和其他许多领域中都有长期的研究基础。在机器智能方面，优化可定义为随着时间学习做出更好的选择，从而使效用函数最大化，最终实现目标(Werbos，2009a)。

为此，在随机系统中，随时间优化的基础是 Von Neumann 所定义的与主要效用函数紧密关联的 Bellman 方程(Bellman，1957)。具体地，给定一个性能代价如下的系统：

$$J[x(i),i]=\sum_{t=i}^{\infty}\gamma^{t-i}U[x(t),u(t),t] \tag{5-1}$$

其中，$x(t)$为系统的状态向量，$u(t)$为控制动作，U 为效用函数，γ 为损耗因子。动态规划的目标是求解控制序列 $u(t)$，以使代价函数 J 最小化：

$$J^*(x(t))=\min_{u(t)}\{U(x(t),u(t))+\gamma J^*(x(t+1))\} \tag{5-2}$$

式(5-2)奠定了逆向时间倒推的动态规划的实现基础。例如，像反向传播神经网络的通用逼近器(Werbos，1983，1988a，1988b，1989，1991，1995)。

现有的自适应评价设计可分为3大类(Prokhorov & Wunsch，1997；Ferrari & Stengel，2004；Werbos，1992)：启发式动态规划(HDP)、双启发式动态规划(DHP)和全局双启发式动态规划(GDHP)。这些都属于第一代 ADP 设计模型(Werbos，1977，2009a)，其目标是通过使用来自评价网络的两个连续估计的时间差分来评价动作值(action value)，从而优化代价函数(Si & Wang，2001)。这种思想本质上类似于 RL 文献(Sutton & Barto，1998)中所讨论的时序差分法(TD)。为了克服可扩展性方面的局限性，Werbos(1979，1981)提出了 DHP 和 GDHP，随后出现了这两种方法的许多改进方法(Si 等，2004；White & Sofge，1992；Werbos，2008)。DHP 的关键思想是使用评价网络输出代价函数相对于状态变量数，而 GDHP 充分利用了 HDP 和 DHP，使用评价网络输出代价函数及其导数。学术界也研究了这些 ADP 设计的变种，如通过以动作值作为评价网络的额外输入，Si & Wang(2001)提出了上述方法的执行依赖(action-dependent，AD)版本。

预测是促进通用智能发展的另一个关键因素(Werbos，2009a)，与第二代 ADP

设计密切相关(Werbos，1987，1992，1993，2009a；Werbos & Pellionisz，1992)。尽管学术界对预测问题已经关注了很长时间，但是从数据挖掘的角度来看，大脑对一般预测过程的理解仍然相当有限。例如，许多现存的预测方案仅仅基于对历史数据模式的观察，而没有理解生物大脑如何在分布式神经组织内跨越不同领域实现鲁棒预测的基本原理。另一方面，神经生物学研究表明，对成年大鼠内侧丘脑腹后(VPM)接受域进行观测，发现73%的神经元对附近区域的刺激表现出即时的短延迟响应(SLR)(Nicolelis，Lin，Woodward & Chapin，1993)，这说明这种细胞可以作为接收到输入信号之前的预测器(Werbos，2009a；Nicolelis，Baccala，Lin & Chapin，1995)。这一研究结果可以支持Werbos(2009a)所强调的预测是类脑通用智能的另一个核心目标的假说。从工程研究的观点来看，预测与优化相结合可以实现更好的性能。例如，大量研究结果表明，使用递归神经网络(RNN)，如细胞同时复发性神经网络(CSRN)与扩展卡尔曼滤波器(EKF)(White & Sofge，1992；Werbos，1999，2004；Ilin，Kozma & Werbos，2008)、时滞递归神经网络(TLRN)(Feldkamp & Prokhorov，2003；Feldkamp，Prokhorov，Eagen & Yuan，1998；Puskorius & Feldkamp，1994；Sun & Marko，1998)相结合，可以有效地对预测和优化进行集成。这反过来又明显提高了神经网络解决复杂问题的能力，诸如复杂场景中的控制和导航(Ilin 等，2008)。这个概念也与 ObjectNet 方法相关联，该方法将复杂输入域映射到不同类型的目标中(“内循环神经网络”)(Werbos，1998a，2009a)，这已被证明在许多复杂问题，如电力网格控制(Qiao，Venayagamoorthy & Harley，2007)、大师级国际象棋程序控制(Fogel，Hays，Han & Quon，2004)中具有巨大的应用潜力。

针对这两个目标，本章基于 ADP 设计提出了一种具有多目标表示的分层学习构架，以实现机器智能研究中的优化和预测。除了行动网络和评价网络的传统 ADP 设计，所提出模型的核心思想是集成另一种网络类型——参考网络，以提供多层次的内部强化表示(类似于次级强化信号的概念)，从而与学习系统相互作用。在提出的 ADP 设计中，参考网络作为一个重要角色去实现目标的优化和预测。不像现有的 ADP 设计使用一个单一的次级强化信号，提出的 ADP 构架使用两种类型的强化信号，即外部环境的初级强化信号和参考网络的次级强化信号，来改进泛化能力和学习能力。初级强化信号可以是二元信号，表示“好”或“坏”/“成功”或“失败”。对于智能系统不同等级的内部目标表示，次级强化信号可以是连续信号。接下来讨论所提出的架构的学习、适应过程及其实际应用。

5.3　机器智能的 ADP

5.3.1　ADP 设计中的分层结构

得益于 ADP 的最新研究进展(Werbos，1992，1998b，2009a；Si & Wang，2001；Si 等，2004；Enns & Si，2004；Si & Liu，2004)、神经生物学研究(Melzack，1990；Peyron，Laurent & Garcia-Larrea，2000；Derbyshire 等，1997；Hsieh，Tu & Chen，2001)，以及分层学习和记忆组织(Starzyk，Liu & He，2006)，本节提出一种具有分层目标表示的 ADP 构架，如图 5-1 所示。这种方法主要是建立一种分层内部强化表示，以提高系统的泛化能力和学习能力。在图 5-1 中，使用多层参考网络自动开发内部强化信号，用分层方式表示不同层次的目标。每个较高层将为低层次的学习提供指导(类似于自顶向下的预测)，最高层接收来自外部环境的初级强化信号，这可视为学习系统的最终目标。基于该初级强化信号自适应地建立多个内部强化信号，从而建立不同层次的目标表示，以促进学习过程。

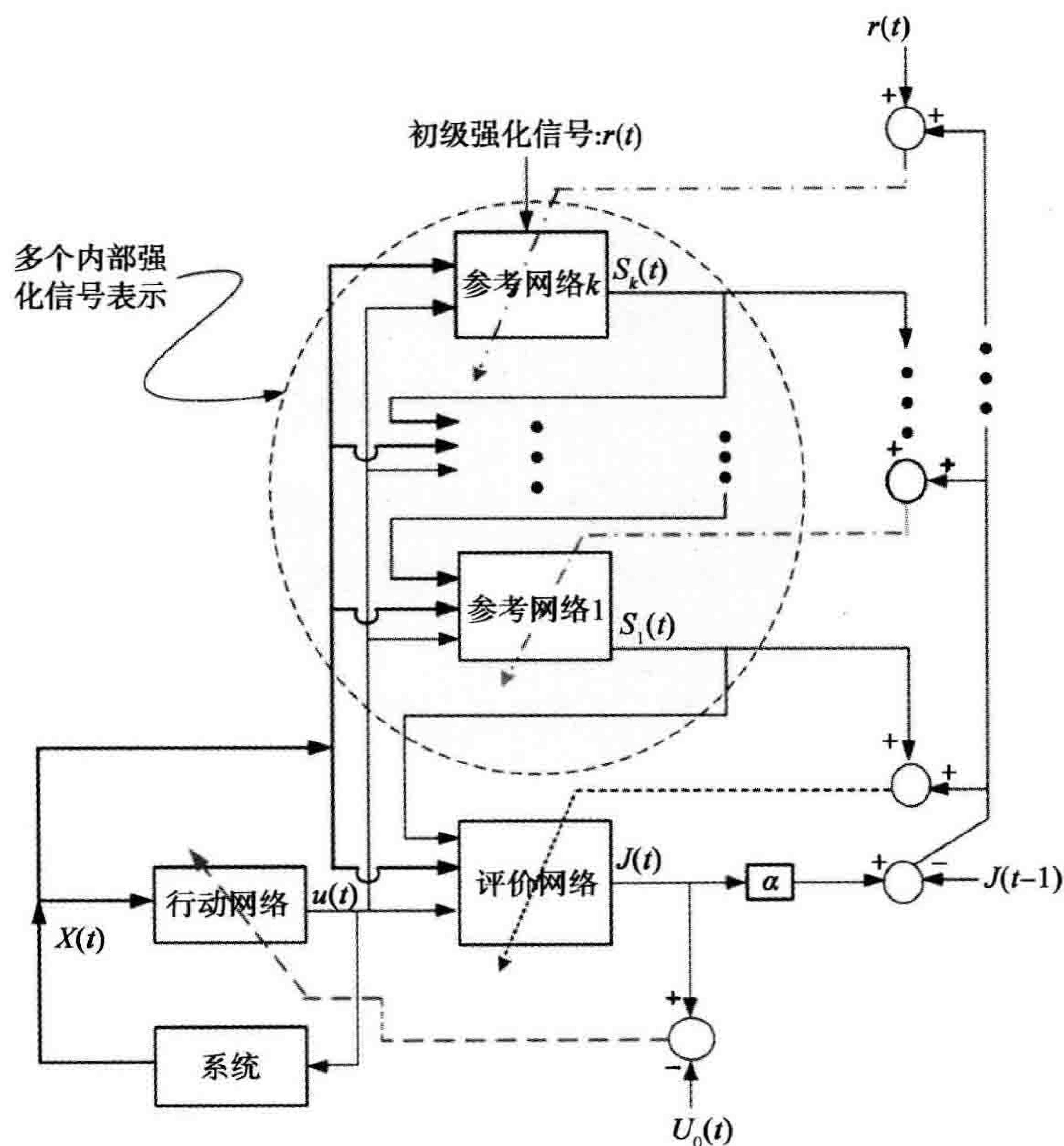


图 5-1　ADP 设计中的多目标强化学习表示

与经典的 ADP 架构设计相比，所提出架构的一个重要特点是结合参考网络来提供多层次的内部强化信号 $s(t)$。这里的内部强化信号类似于 Werbos(2009a)和 Frijida(1987)讨论的“次级强化信号”，在建立参考网络与评价网络之间的相互作用方面扮演着重要角色。值得注意的是，参考网络的使用也与第一代和第二代 ADP 设计(Werbos，2009a)中的“模型网络”类似。例如，在第一代 ADP 设计中，模型网络被用来学习如何预测环境的变化并估计目标的实际状态(Werbos，2009a)。在第二代 ADP 设计中，模型网络具有更强的预测能力，这可以通过递归网络实现(也被称为“在一个模型中的两个大脑”，具有高低水平的适应性评价系统 Werbos，(2009a)。在提出的 ADP 构架中，参考网络的目标是为学习和优化过程提供预测能力。因此，在该架构中提出的参考网络也可以被视为 Werbos(2009a)的网络模型中的一个特定设计策略。

提出的 ADP 体系架构包括两种类型的强化信号，观察和分析这两种强化表示如何促进学习过程是非常有趣的。在传统的 ADP 设计中，强化信号通常被认为是一个二元信号，如分别使用 0 和 −1 代表系统的“成功”和“失败”(Si 和 Liu，2004)。在许多复杂问题中，为了提供信息丰富的强化信号，有很多方法使用非二元强化信号来提高 ADP 设计的学习性能。例如，在 Si 和 Wang(2001)，除了传统的二元强化表示之外，还提出了一种不同的设计策略，即利用一个三值强化信号(0，−0.4 和 −1)表示钟摆摆动和平衡。最近，Enns & Si(2004) 提出一种关于 ADP 设计的次级强化信号，它为每个采样时间提供更丰富的信息：

$$r(t)=-\sum_{i=1}^{n}\left(\frac{(x_i-x_{i,d})}{x_{i,\max}}\right)^2 \tag{5-3}$$

其中，x_i 为状态向量 x 的第 i 种状态，$x_{i,d}$ 为期望的参考状态，$x_{i,\max}$ 为形式上的最大状态值。通过这种方式，ADP 构架表现出较好的泛化和学习性能，并成功地应用于直升机的飞行控制中(Enns & Si，2004)。

本章提出的构架如图 5-1 所示，内部强化信号 $s(t)$ 是用一种更自然和富有原则性的方式建立的。也就是说，这样的内部强化表示是通过参考网络的不同层次自动建立的，它反过来与评价网络合作，从而进行优化和学习。

另一个使用信息丰富的内部强化表示的有力支持来自心理学研究。心理学研究认为，生物系统能够开发内部目标表示，以触发具身智能的感觉与运动传导路径之间的学习和联想。例如，使用正电子发射断层成像(PET)和功能性磁共振成像(fMRI)技术的神经生物学研究表明，疼痛系统涉及大脑的多个区域，由这些区域形成了痛感矩阵(Neomatrix)，以表示内在的动机/价值信号表示(Melzack，1990；

Peyron 等，2000；Derbyshire 等，1997；Hsieh 等，2001)。因此，含有多层强化信号的目标创造系统的层次化表示(Starzyk 等，2006) 在理解大脑功能方面提供了对优化和预测的深入理解。

本章还讨论如何实现预测并将其集成到 ADP 架构中。一般来说，在 ADP 设计中，预测可以被认为是以一种更一般的方法包含了更多重要信息，如来自观察数据的感觉输入，以及对未观察到的状态变量的建模和重构，具有促进动作选择从而实现优化的目标(Werbos，2009a)。因此，在所提出的 ADP 架构中，预测能力是通过分层地组织参考网络以预测内部强化信号 $s(t)$，从而为改进优化性能提供信息量丰富的目标表示来实现的。对 $s(t)$信号的预测，可以通过诸如神经网络的非线性函数逼近器实现(包括递归网络)。现在继续讨论下列 3 种类型网络的详细学习和适应过程：行动网路、评价网络和参考网络。

5.3.2 ADP 设计中的学习和自适应

为了清楚地描述 ADP 构架的学习机制，我们以一种仅具有参考网络的特定双层构架为例来讨论它的学习和自适应过程，如图 5-2 所示。其中，使用了反向传播规则，并有 3 条参数转化路径，以调整 3 种类型网络的参数。

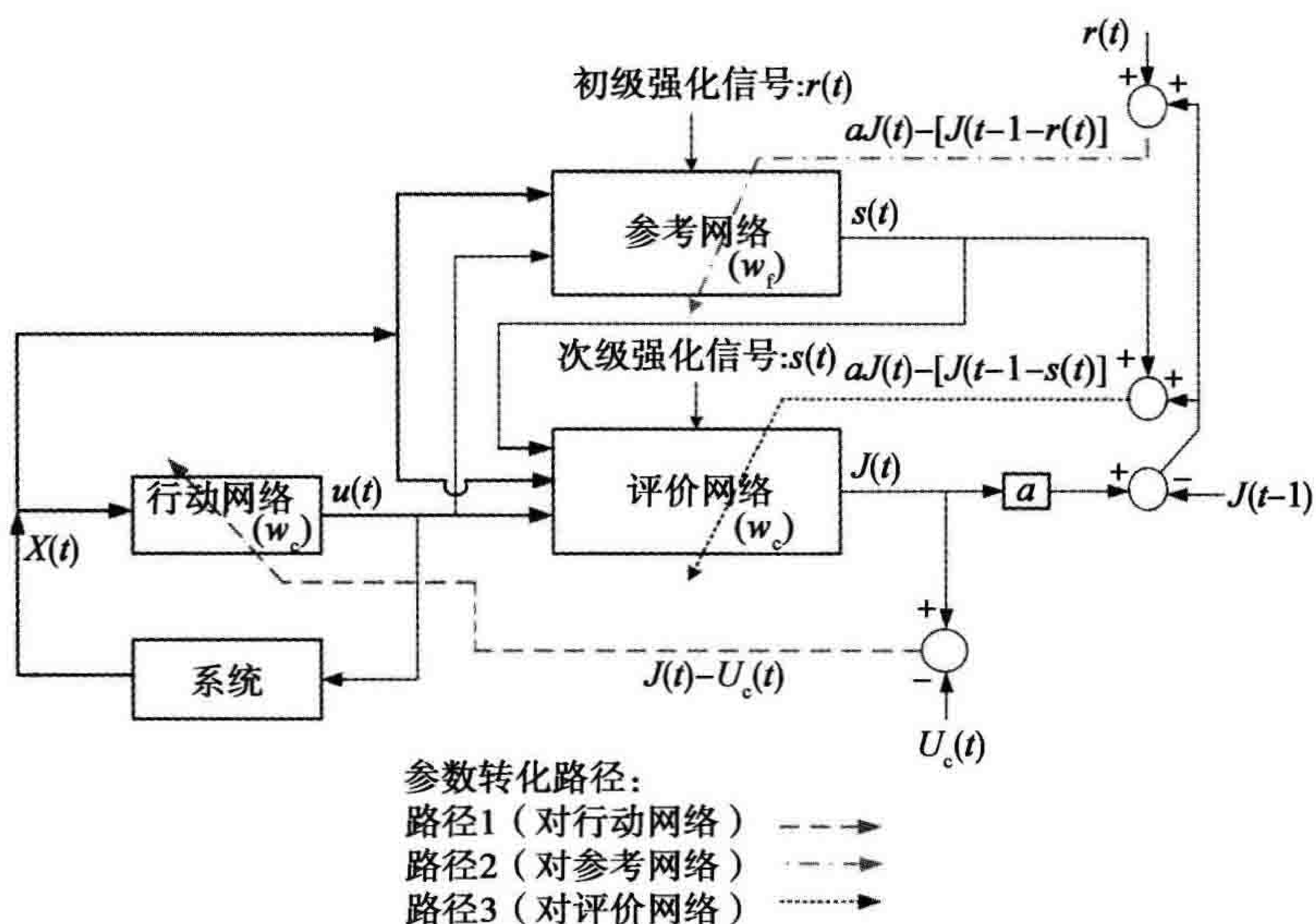


图 5-2　具有 3 种网络的 ADP 构架

这种结构的行动网络类似于经典的 ADP 方法，能间接地反向传播期望目标 U_c 与来自评判网络的 J 函数之间的误差(Si & Wang，2001)(Si & Liu，2004)。因此，

它可用于更新行动网络参数的误差函数 $E_a(t)$，具体定义如下(见图 5-2 中的路径 1)：

$$e_a(t) = J(t) - U_c(t);\ E_a(t) = \frac{1}{2}e_a^2(t) \tag{5-4}$$

这种架构主要依赖于参考网络和评价网络的学习与自适应过程。由于初级强化信号 $r(t)$出现在参考网络中，次级强化信号 $s(t)$自适应地向评价网络提供信息丰富的内部强化表示，$s(t)$反过来也是对 $J(t)$的一个更好的近似。这样，初级强化信号 $r(t)$位于较高层次，可以用一个简单的二元信号代表“好”或“坏”(“成功”或“失败”)，而次级强化信号 $s(t)$可以是一个信息量更丰富的连续值，用于提高学习和泛化性能。因此，用于更新参考网络参数的误差函数 $E_f(t)$可以被定义为(见图 5-2 中的路径 2)：

$$e_f(t) = \alpha J(t) - [J(t-1) - r(t)];\ E_f(t) = \frac{1}{2}e_f^2(t) \tag{5-5}$$

一旦参考网络输出 $s(t)$信号，它将作为评价网络的输入，并用于定义调整评价网络参数的误差函数(见图 5-2 中的路径 3)：

$$e_c(t) = \alpha J(t) - [J(t-1) - s(t)];\ E_c(t) = \frac{1}{2}e_c^2(t) \tag{5-6}$$

从数学的角度来看，这一框架和传统的评价网络设计相比，有两个主要不同点。首先，评判网络具有一个来自参考网络的额外输入 $s(t)$；第二，参考网络与评价网络的优化误差函数是不同的——参考网络的误差函数与初级强化信号 $r(t)$相关(见式(5-5))，而评价网络的误差函数与次级强化信号 $s(t)$相关(见式(5-6))。主要思想是使用这样的一个次级强化信号通过关联和预测构建 ADP 的内部目标表示。

在这种构架中，链式反向传播规则对这 3 种网络(行动网络、评价网络和参考网络)的参数训练和调整起着关键作用(Werbos，1990，1994)。图 5-3 展示了 3 条用于调整网络中的参数的反向传播路径。其中，行动网络 E_a、参考网络 E_f 和评价网络 E_c 的优化误差函数分别如式(5-4)、式(5-5)和式(5-6)所示。因此，链式反向传播可以通过 3 条数据路径计算，如图 5-3 所示。简单地说，3 条路径的计算可概括如下。

路径 1 行动网络：

$$\frac{\partial E_a(t)}{\partial w_a(t)} = \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial w_a(t)} \tag{5-7}$$

路径 2 参考网络：

$$\frac{\partial E_f(t)}{\partial w_f(t)} = \frac{\partial E_f(t)}{\partial J(t)} \frac{\partial J(t)}{\partial s(t)} \frac{\partial s(t)}{\partial w_f(t)} \tag{5-8}$$

路径 3 评价网络：

$$\frac{\partial E_c(t)}{\partial w_c(t)} = \frac{\partial E_c(t)}{\partial J(t)} \frac{\partial J(t)}{\partial w_c(t)} \tag{5-9}$$

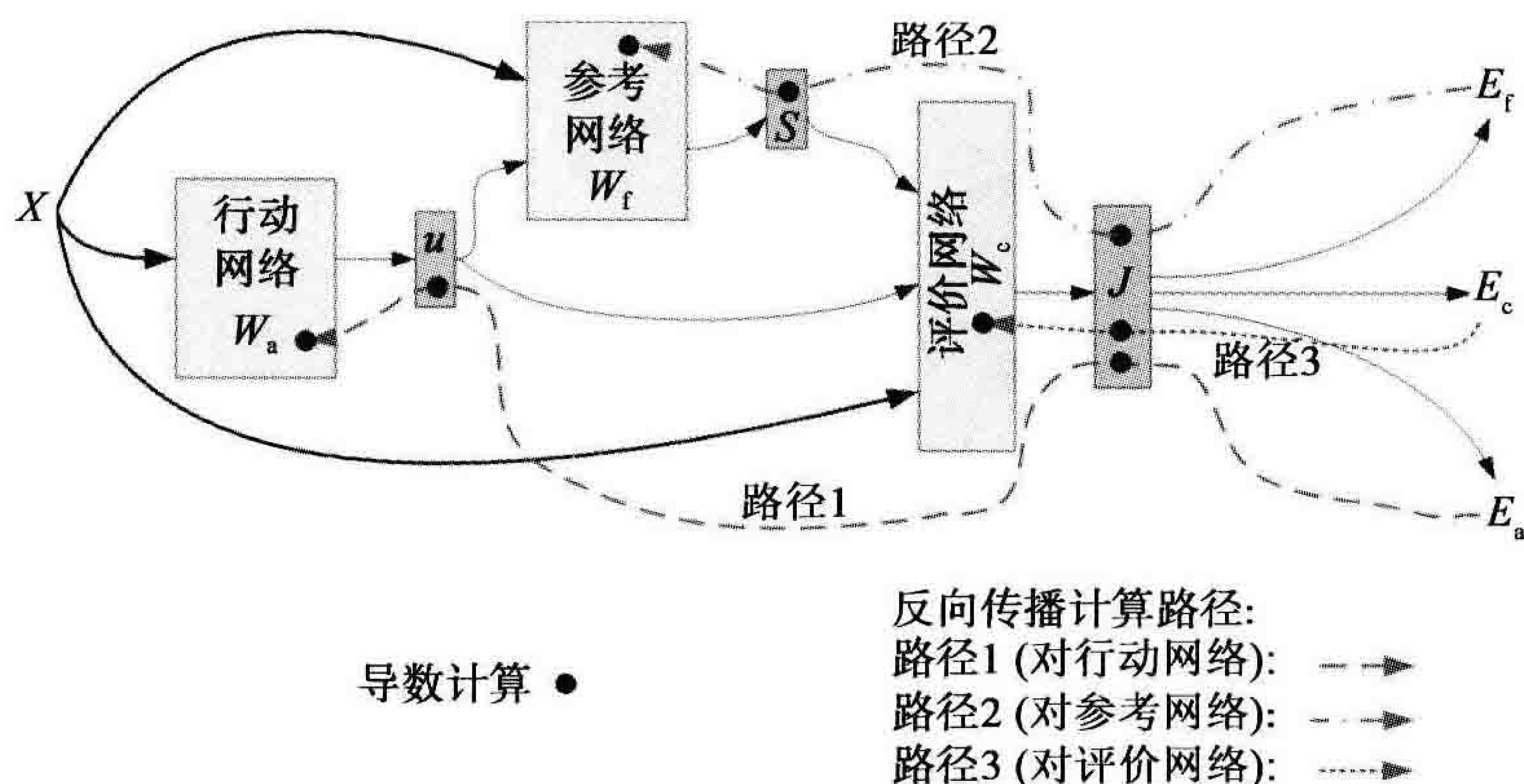


图 5-3　基于反向传播的自适应和参数调整

现在，详细讨论所提出的 ADP 构架中 3 种网络的学习和自适应过程。

1. 行动网络

为了强调学习原理，本章假设在 3 种网络中均使用具有 3 层非线性结构的神经网络(含有一个隐层)。值得注意的是，通过适当地应用反向传播规则，这里所讨论的学习原理可以推广到任意的函数逼近器。

图 5-4 显示了该设计中使用的行动网络，包括系统状态 X 的 n 个输入和动作值 u 的一个输出。调整行动网络的原则是间接地反向传播所期望的最终目标 $U_c(t)$ 与来自评价网络的近似 J 函数之间的误差。类似于 Si & Wang(2001)以及 Si & Liu(2004)，由于在初级强化信号中定义“0”表示“成功”，所以在当前设计中 $U_c(t)$ 定义为“0”。正如在 5.3.2 节中所讨论的，误差函数 $E_a(t)$ 用于更新行动网络的参数，如式(5-4)所示。根据图 5-4，行动网络的相关方程可被定义为

$$u(t)=\frac{1-\exp^{-v(t)}}{1+\exp^{-v(t)}} \tag{5-10}$$

$$v(t)=\sum_{i=1}^{N_h} w_{a_i}^{(2)}(t)g_i(t) \tag{5-11}$$

$$g_i(t)=\frac{1-\exp^{-h_i(t)}}{1+\exp^{-h_i(t)}},\ i=1,\cdots,N_h \tag{5-12}$$

$$h_i(t)=\sum_{j=1}^{n} w_{a_i,j}^{(1)}(t)x_j(t),\ i=1,\cdots,N_h \tag{5-13}$$

其中，h_i 为行动网络第 i 个隐藏节点的输入，g_i 为对应的输出，v 为 Sigmoid 函数输出节点的输入，N_h 为行动网络隐藏神经元的数量，n 为行动网络的输入总数量(见图 5-4)。

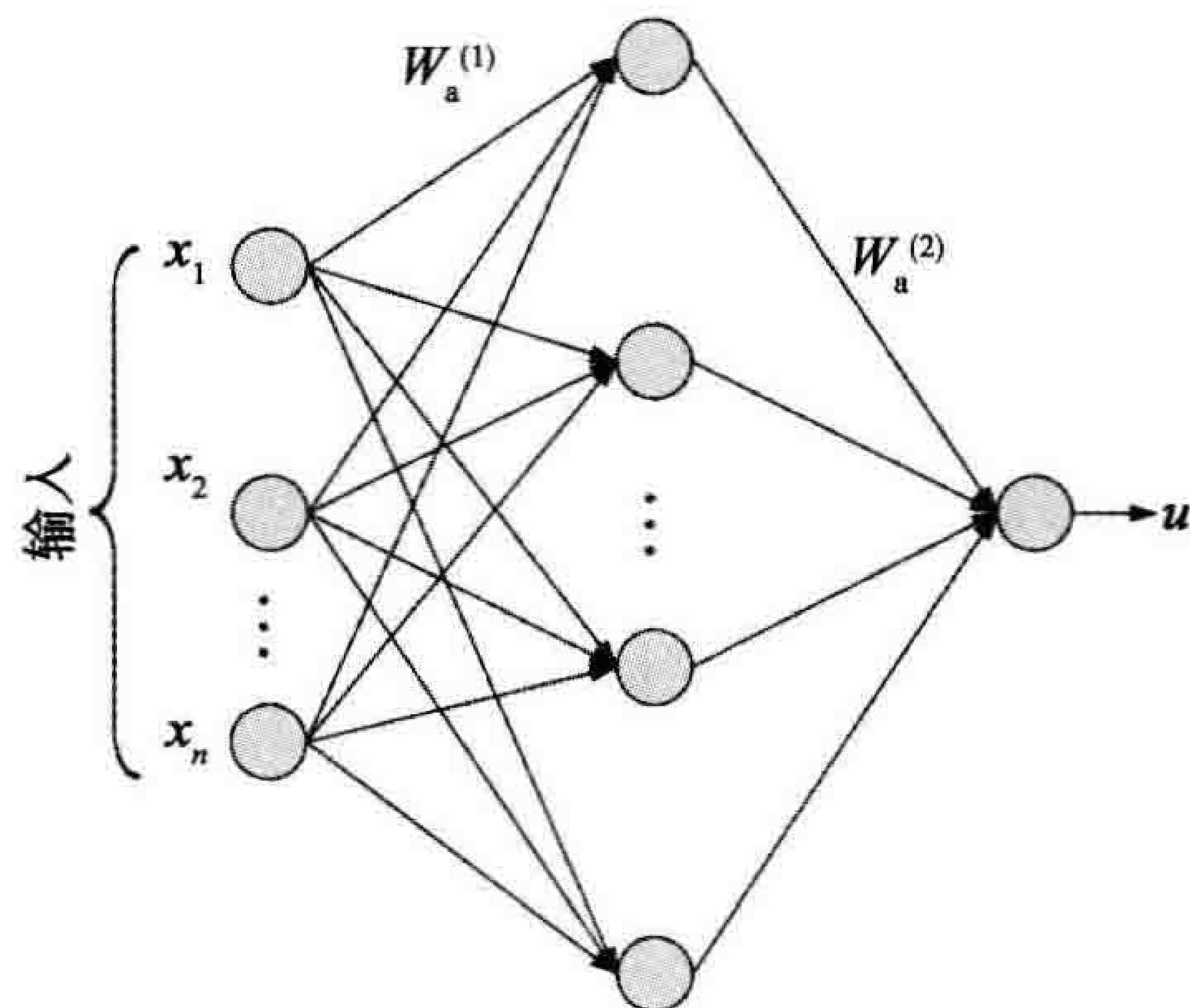


图 5-4 具有一个隐层的非线性神经网络的行动网络

类似于经典的 ADP 设计(Si & Wang, 2001; Si & Liu, 2004)，应用链式反向传播规则可以实现行动网络的自适应，具体描述如下。

1) $\Delta w_a^{(2)}$：在行动网络中，隐层到输出层的权重调整如下。

$$\Delta w_{a_i}^{(2)} = \eta_a(t)\left[-\frac{\partial E_a(t)}{\partial w_{a_i}^{(2)}(t)}\right] \tag{5-14}$$

$$\begin{aligned}\frac{\partial E_a(t)}{\partial w_{a_i}^{(2)}(t)} &= \frac{\partial E_a(t)}{\partial J(t)}\frac{\partial J(t)}{\partial u(t)}\frac{\partial u(t)}{\partial v(t)}\frac{\partial v(t)}{\partial w_{a_i}^{(2)}(t)} \\ &= e_a(t)\cdot\sum_{i=1}^{N_h}\left[w_{c_i}^{(2)}(t)\frac{1}{2}(1-p_i^2(t))w_{c_i,n+1}^{(1)}(t)\right] \\ &\quad\cdot\frac{1}{2}(1-(u(t))^2).\, g_i(t)\end{aligned} \tag{5-15}$$

2) $\Delta w_a^{(1)}$：在行动网络中，输入层到隐层的权重调整如下。

$$\Delta w_{a_{i,j}}^{(1)} = \eta_a-(t)\left[-\frac{\partial E_a(t)}{\partial w_{a_{i,j}}^{(1)}(t)}\right] \tag{5-16}$$

$$\begin{aligned}\frac{\partial E_a(t)}{\partial w_{a_{i,j}}^{(1)}(t)} &= \frac{\partial E_a(t)}{\partial J(t)}\frac{\partial J(t)}{\partial u(t)}\frac{\partial u(t)}{\partial v(t)}\frac{\partial v(t)}{\partial g_i(t)}\frac{\partial g_i(t)}{\partial h_i(t)}\frac{\partial h_i(t)}{\partial w_{a_{i,j}}^{(1)}(t)} \\ &= e_a(t).\sum_{i=1}^{N_h}\left[w_{c_i}^{(2)}(t)\frac{1}{2}(1-p_i^2(t))w_{c_i,n+1}^{(1)}(t)\right] \\ &\quad\cdot\frac{1}{2}(1-(u(t))^2)\cdot w_{a_i}^{(2)}(t)\cdot\frac{1}{2}(1-g_{g_i}^2(t))\cdot x_j(t)\end{aligned} \tag{5-17}$$

2. 参考网络

图 5-5 显示了该设计中使用的 3 层非线性架构参考网络(具有一个隐层)，为了计算反向传播，首先定义参考网络的输出 $s(t)$，描述如下：

$$s(t) = \frac{1-\exp^{-k(t)}}{1+\exp^{-k(t)}} \tag{5-18}$$

$$k(t)=\sum_{j=1}^{N_h} w_{f_i}^{(2)}(t) y_j(t) \tag{5-19}$$

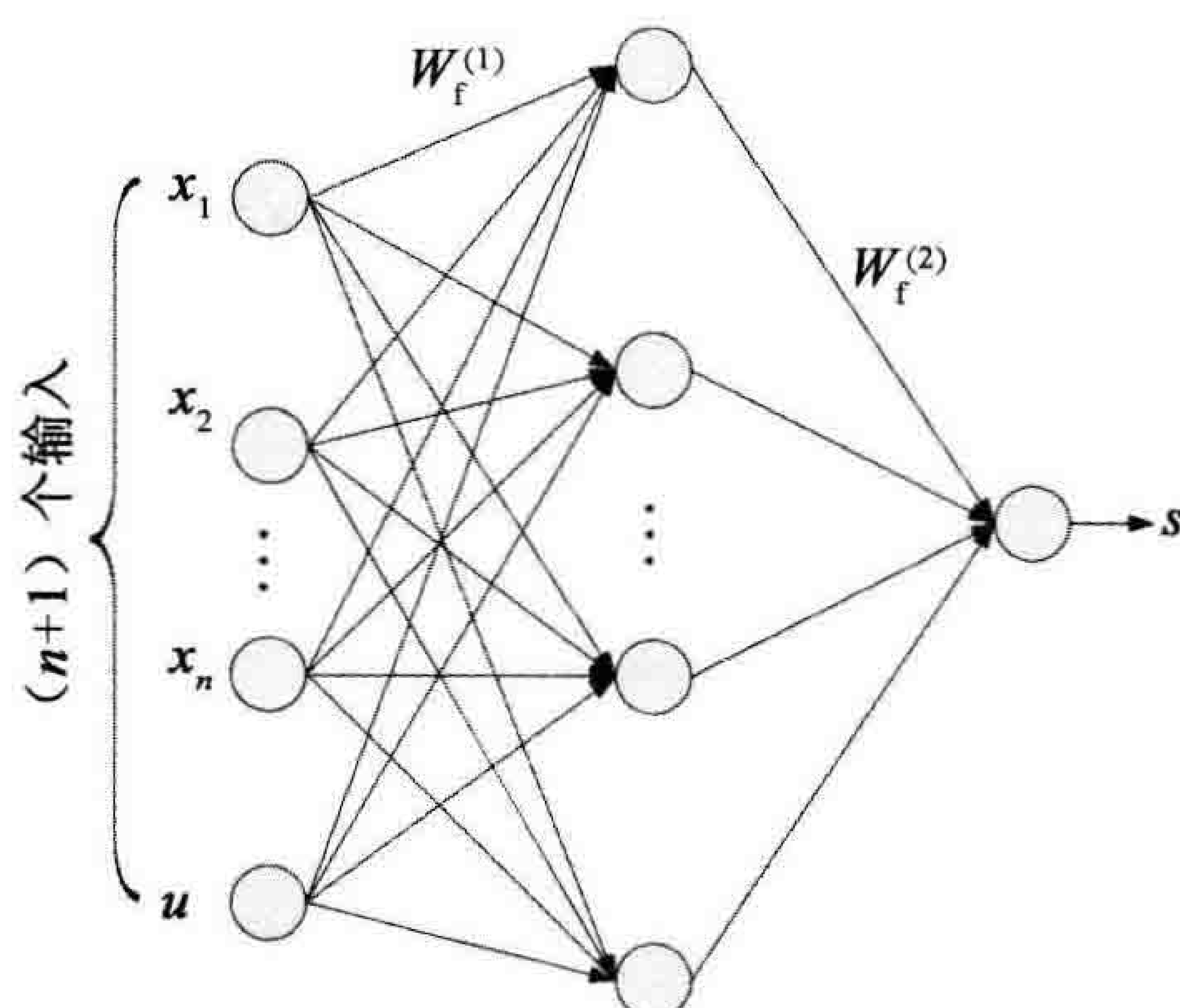


图 5-5 具有一个隐层的非线性神经网络的参考网络

$$y_j(t)=\frac{1-\exp^{-z_i(t)}}{1+\exp^{-z_i(t)}},\ i=1,\cdots,N_h \tag{5-20}$$

$$z_i(t)=\sum_{j=1}^{n+1} w_{f_{i,j}}^{(1)}(t) x_j(t),\ i=1,\cdots,N_h \tag{5-21}$$

其中，z_i 为参考网络的第 i 个隐藏节点的输入，y_i 为对应的输出，k 为 Sigmoid 函数的输出节点的输入，N_h 为参考网络隐藏神经元的数量，$(n+1)$为参考网络的输入总数量，包括来自行动网络的动作值 $u(t)$(见图 5-4)。

为了应用反向传播规则，可以参考图 5-3 和式(5-5)、式(5-8)。尤其是，由于输出 $s(t)$是评价网络的输入，通过链式规则(路径 2)可以应用反向传播调整参数 W_f，该过程的描述如下。

1) $\Delta w_f^{(2)}$：在参考网络中，隐层到输出层的权重调整如下。

$$\Delta w_{f_i}^{(2)}=\eta_f(t)\left[-\frac{\partial E_f(t)}{\partial w_{f_i}^{(2)}(t)}\right] \tag{5-22}$$

$$\begin{aligned}\frac{\partial E_f(t)}{\partial w_{f_i}^{(2)}(t)} &=\frac{\partial E_f(t)}{\partial J(t)}\frac{\partial J(t)}{\partial s(t)}\frac{\partial s(t)}{\partial k(t)}\frac{\partial k(t)}{\partial w_{f_i}^{(2)}(t)} \\ &=\alpha e_f(t)\cdot\sum_{i=1}^{N_h}\left[w_{c_i}^{(2)}(t)\frac{1}{2}(1-p_i^2(t))w_{c_i,n+2}^{(1)}(t)\right] \\ &\quad\cdot\frac{1}{2}(1-(s(t))^2)\cdot y_i(t)\end{aligned} \tag{5-23}$$

2) $\Delta w_f^{(1)}$：在参考网络中，输入层到隐层的权重调整如下。

$$\Delta w_{\mathrm{f}_{i,j}}^{(1)} = \eta_{\mathrm{f}}(t)\left[-\frac{\partial E_{\mathrm{f}}(t)}{\partial w_{\mathrm{f}_{i,j}}^{(1)}(t)}\right] \tag{5-24}$$

$$\begin{aligned}\frac{\partial E_{\mathrm{f}}(t)}{\partial w_{\mathrm{f}_{i,j}}^{(1)}(t)} &= \frac{\partial E_{\mathrm{f}}(t)}{\partial J(t)}\frac{\partial J(t)}{\partial s(t)}\frac{\partial s(t)}{\partial k(t)}\frac{\partial k(t)}{\partial y_i(t)}\frac{\partial y_i(t)}{\partial z_i(t)}\frac{\partial z_i(t)}{\partial w_{\mathrm{f}_{i,j}}^{(1)}(t)} \\ &= \alpha e_{\mathrm{f}}(t)\cdot\sum_{i=1}^{N_{\mathrm{h}}}\left[w_{\mathrm{c}_i}^{(2)}(t)\,\frac{1}{2}(1-p_i^2(t))w_{\mathrm{c}_i,n+2}^{(1)}(t)\right] \\ &\quad\cdot\frac{1}{2}(1-(s(t))^2)\cdot w_{\mathrm{f}_i}^{(2)}(t)\cdot\frac{1}{2}(1-y_i^2(t))\cdot x_j(t)\end{aligned} \tag{5-25}$$

一旦参考网络为评判网络提供了次级强化信号 $s(t)$，便可以调整评价网络中的参数。

3. 评价网络

图 5-6 显示了本章的设计中使用的 3 层非线性架构评价网络(具有一个隐层)，为了计算反向传播，首先定义评价网络的输出 $J(t)$，描述如下：

$$J(t) = \sum_{i=1}^{N_{\mathrm{h}}} w_{\mathrm{c}_i}^{(2)}(t)p_i(t) \tag{5-26}$$

$$p_i(t) = \frac{1-\exp^{-q_i(t)}}{1+\exp^{-q_i(t)}},\ i=1,\cdots,N_{\mathrm{h}} \tag{5-27}$$

$$q_i(t) = \sum_{j=1}^{n+2} w_{\mathrm{c}_{i,j}}^{(1)}(t)x_j(t),\ i=1,\cdots,N_{\mathrm{h}} \tag{5-28}$$

其中，q_i 和 p_i 分别是评价网络第 i 个隐藏节点的输入和输出，$(n+2)$是评价网络的输入总数量，包括行动网络的动作值 $u(t)$和参考网络的次级强化信号 $s(t)$。

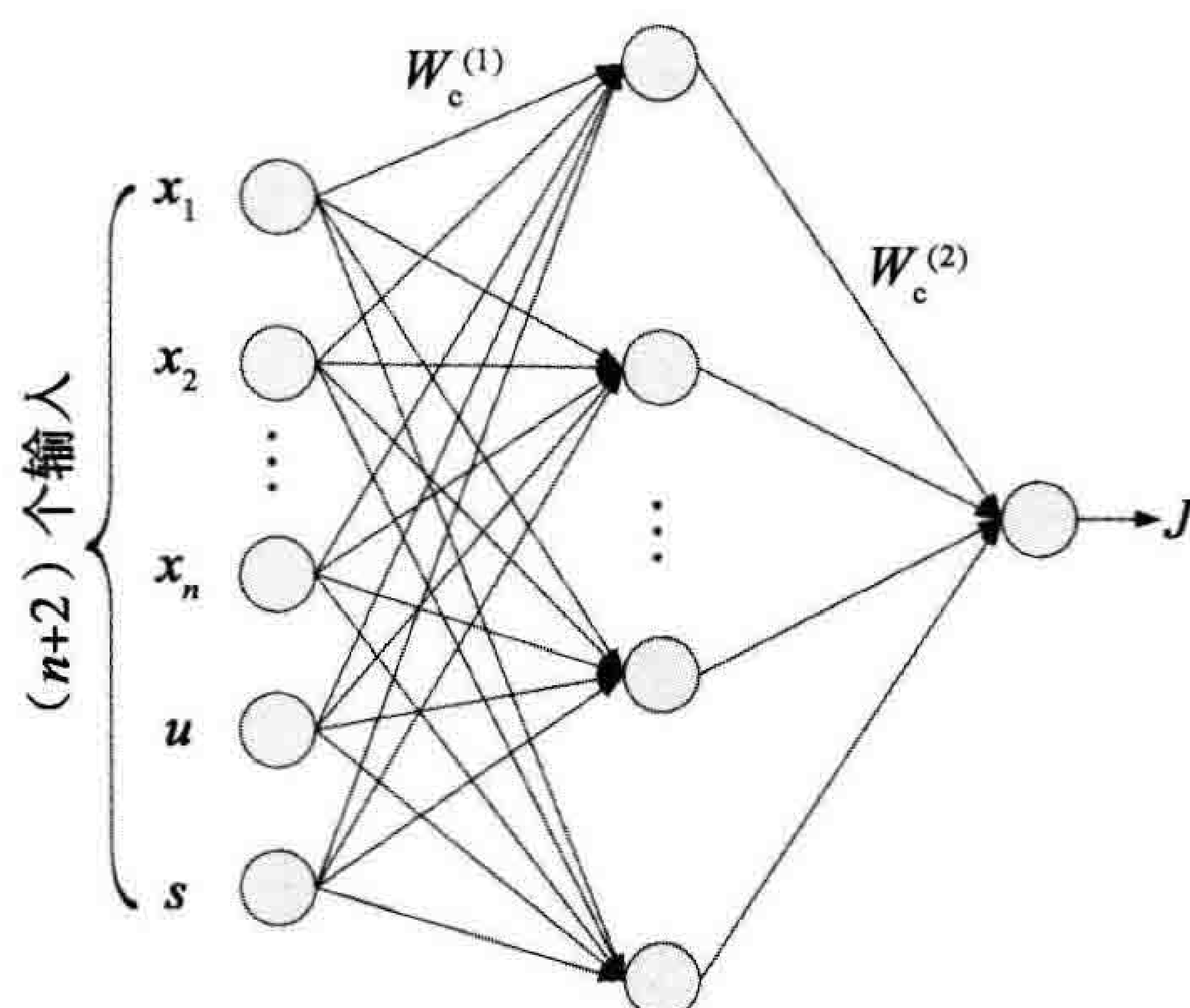


图 5-6　具有一个隐层的非线性神经网络的评价网络

通过链式反向传播规则(路径 3)，评价网络中参数调整的过程概括如下。

1) $\Delta w_{\mathrm{c}}^{(2)}$：在评价网络中，隐层到输出层的权重调整如下。

$$\Delta w_{\mathrm{c}_i}^{(2)} = \eta_{\mathrm{c}}(t)\left[-\frac{\partial E_{\mathrm{c}}(t)}{\partial w_{\mathrm{c}_i}^{(2)}(t)}\right] \tag{5-29}$$

$$\frac{\partial E_c(t)}{\partial w_{c_i}^{(2)}(t)}=\frac{\partial E_c(t)}{\partial J(t)}\frac{\partial J(t)}{\partial w_{c_i}^{(2)}(t)}=\alpha e_c(t)\cdot p_i(t) \tag{5-30}$$

2) $\Delta w_c^{(1)}$：在评价网络中，输入层到隐层的权重调整如下。

$$\Delta w_{c_{i,j}}^{(1)}=\eta_c(t)\left[-\frac{\partial E_c(t)}{\partial w_{c_{i,j}}^{(1)}(t)}\right] \tag{5-31}$$

$$\begin{aligned}\frac{\partial E_c(t)}{\partial w_{c_{i,j}}^{(1)}(t)}&=\frac{\partial E_c(t)}{\partial J(t)}\frac{\partial J(t)}{\partial p_i(t)}\frac{\partial p_i(t)}{\partial q_i(t)}\frac{\partial q_i(t)}{\partial w_{c_{i,j}}^{(1)}(t)}\\&=\alpha e_c(t)\cdot w_{c_i}^{(2)}(t)\cdot\frac{1}{2}(1-p_i^2(t))\cdot x_j(t)\end{aligned} \tag{5-32}$$

5.3.3　学习策略：序贯学习和协同学习

观察图 5-2 和图 5-3 可以看出，$s(t)$信号提供了参考网络与评价网络之间的重要链接，这使得链式反向传播能够在参考网络和评价网络中调整参数。可以看出，在实践中可以执行两种学习策略。

首先是序贯学习，如图 5-7a 所示。在这种学习策略中，首先根据式(5-22)～式(5-25) 调整参考网络 W_f 的权重值；其次，参考网络输出信号 $s(t)$，根据式(5-29)～式(5-32)调整评价网络；最后根据式(5-14)～式(5-17)，可以调整行动网络的权重值。

第二种是协同学习，如图 5-7b 所示，更多涉及参考网络和评价网络之间的交互。在此学习策略的每个阶段，首先通过反向传播调整基于初级强化信号 $r(t)$的参考网络 W_f。然后参考网络输出次级强化信号 $s(t)$，其通过反向传播调整评价网络 W_c 的参数。一旦 W_c 在该阶段被调整，评价网络将提供一个新的 $J(t)$估计，反过来在下一个阶段中可用于调整 W_f。这样，参考网络和评价网络以更协作的方式进行训练。

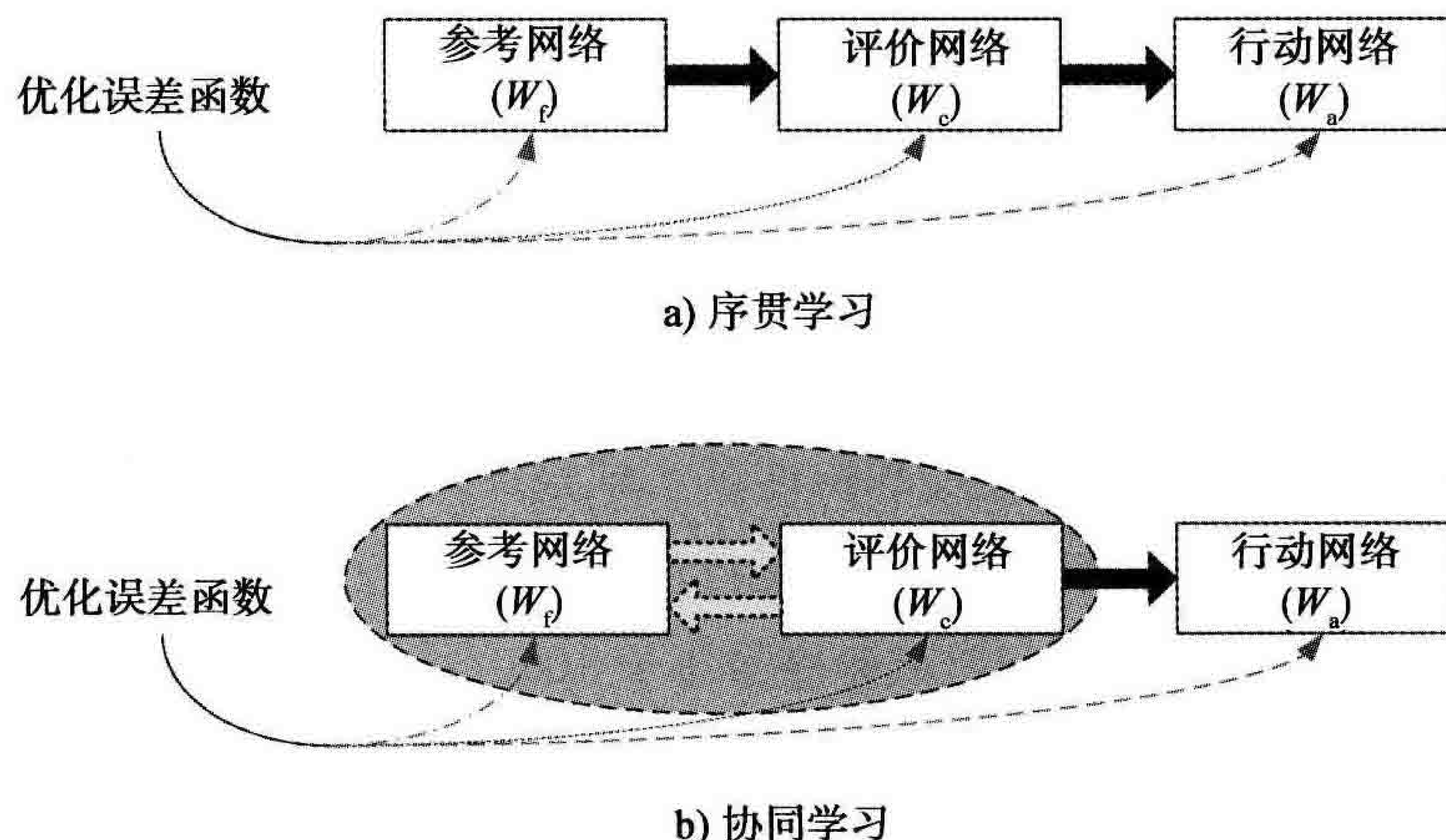


图 5-7　ADP 构架的两种学习策略

5.4 实例研究

本节给出了上述 ADP 构架控制倒立摆平衡的实验结果。倒立摆是本领域的一种通用测试问题(Si & Wang，2001；Si & Liu，2004；Anderson，1987，1989；Barto，Sutton & Anderson，1983)(Moriarty & Miikulainen，1996)。实验中的倒立摆平衡模型与 Si & Wang(2001)、Si & Liu(2004)、Barto 等(1983)所描述的完全相同，可以表示如下：

$$\frac{d^2\theta}{dt^2}=\frac{g\sin\theta+\cos\theta\left[-F-ml\dot{\theta}^2\sin\theta+\mu_c\operatorname{sgn}(\dot{x})\right]-\dfrac{\mu_p\dot{\theta}}{ml}}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2\theta}{m_c+m}\right)} \tag{5-33}$$

$$\frac{d^2x}{dt^2}=\frac{F+ml\left[\dot{\theta}^2\sin\theta-\ddot{\theta}\cos\theta\right]-\mu_c\operatorname{sgn}(\dot{x})}{m_c+m} \tag{5-34}$$

其中，

g：9.8m/s^2，重力加速度；

m_c：1.0kg，小车质量；

m：0.1kg，摆杆质量；

l：0.5m，摆杆半长；

μ_c：0.000 5，小车与轨道之间的摩擦系数；

μ_p：0.000 002，摆杆与小车之间的摩擦系数；

F：±10N，小车质心所受的力；

$$\operatorname{sgn}(x):\operatorname{sgn}(x)=\begin{cases}1, & x>0\\ 0, & x=0\\ -1, & x<0\end{cases}$$

该系统有 4 个状态变量：$x(t)$为小车在轨道上的位置，$\theta(t)$为摆杆与垂直方向的夹角，$\dot{x}$ 和 $\dot{\theta}$ 分别为小车的速度和角速度。试验设置也与 Si 和 Wang(2001)中描述的相似。特别是，仿真实验中的一次运行包含最多 1000 次连续试验。一次成功的运行是指该运行的最后一次试验(试验序号小于 1000)持续 60 万个时间步长。否则，如果控制器不能在 1000 次试验中学会平衡，也就是说，在一次运行里的 1000 次试验中没有一次试验可以持续超过 60 万个时间步长，则该运行就是不成功的。每次运行的初始化条件随机设定，定义一次试验为摆杆从开始摆动到落下的完整过程(采用

与 Si 和 Wang(2001)中相同的时间步长 0.02)。当摆杆的倾斜度超过[-12^0，12^0]或者小车运动范围相对于轨迹中心位置超过[−2.4，+2.4]m，该摆杆被认为是落下的。在这个实验中，初级强化信号 $r(t)$用一个简单的二元变量，0 表示“成功”，−1 表示“失败”。对于次级强化信号 $s(t)$，所提出的 ADP 构架中的参考网络(见图 5-2)会自动建立一个内部强化表示，以促进学习和优化过程。

表 5-1 总结了本章所使用的所有实验参数。为了公平比较，采用了与 Si 和 Wang(2001)中所描述的行动网络、评判网络相同的参数。对于行动网络、评价网络和参考网络，它们的权重值 w_a、w_c 和 w_f 是根据其对应的内部周期 N_a、N_c 和 N_f 调整的。这意味着在每个时间步长之内，其各自的权重最多分别更新 N_a、N_c 和 N_f 次，或者，一旦满足其对应的内部训练误差阈值 T_a、T_c 和 T_f，则停止更新。这与 Si 和 Wang(2001)中的构架是相同的。

表 5-1　参数汇总

参数符号	参数表示	参数值
$\eta_c(0)$	评价网络初始学习率	0.3
$\eta_a(0)$	行动网络初始学习率	0.3
$\eta_f(0)$	参考网络初始学习率	0.3
$\eta_c(t)$	t 时刻评价网络学习率	每隔 5 个时间步长，$\eta_c(t)$下降 0.05，直到达到 0.005，此后保持不变
$\eta_a(t)$	t 时刻行动网络学习率	每隔 5 个时间步长，$\eta_a(t)$下降 0.05，直到达到 0.005，此后保持不变
$\eta_f(t)$	t 时刻参考网络学习率	每隔 5 个时间步长，$\eta_f(t)$下降 0.05，直到达到 0.005，此后保持不变
N_c	评价网络内部反向传播周期	50
N_a	行动网络内部反向传播周期	100
N_f	参考网络内部反向传播周期	50
T_c	评价网络内部培训误差阈值	0.05
T_a	行动网络内部培训误差阈值	0.005
T_f	参考网络内部培训误差阈值	0.05
N_h	隐藏节点个数	6

此外，为了评价不同噪声条件下的控制性能，与 Si 和 Wang(2001)的噪声实验相似，本实例也用两种传感器和执行机构进行了一系列实验。执行机构的噪声添加过程具体如下：

$$u(t)=u(t)\times(1+\rho) \tag{5-35}$$

其中，ρ 是噪声率，对于执行机构仅仅考虑均匀噪声。关于传感器噪声，可用下式添加噪声：

$$\theta=\theta\times(1+\rho) \tag{5-36}$$

这里，ρ 是噪声率，在此只考虑均匀噪声和高斯噪声(具有零均值和特定的方差)。

表 5-2 给出了所提出的 ADP 架构的仿真结果，所有结果都是对 Si 和 Wang (2001)中的实验的 100 次随机运行结果的平均值。在这里，成功率被定义为成功运行的次数相对于总运行次数的比率(该实例中总运行次数为 100 次)，试验次数是所有成功运行中试验次数的平均值。表 5-2 显示，在不同的噪声条件下，这两种方法都可以达到 100%的成功率。通过观察表 5-2 最后两例中的试验平均数可以看出，本章提出的 ADP 架构能够提供更好的性能。图 5-8a 和图 5-8b 展示了一个典型的位置轨迹(以米为单位)及其相应成功运行的直方图，图 5-9a 和图 5-9b 展示了一个典型的角度轨迹(以度为单位)及其相应成功运行的直方图。这些直方图说明了倒立摆系统的位置和角度的变化，这些图都是在无噪声情况下得到的。从这些结果可以看出，本章提出的 ADP 构架在该实例研究中具有明显的优越性。

表 5-2 本章的方法与 Si 和 Wang(2001)方法的性能比较

噪声类型	成功率		实验数量	
	参考(Si & Wang，2001)	ADP 构架	参考(Si & Wang，2001)	ADP 构架
无噪声	100%	100%	6	**5.5**
5%的均匀噪声，执行机构	100%	100%	8	**6.86**
10%的均匀噪声，执行机构	100%	100%	14	**9.33**
5%的均匀噪声，传感器	100%	100%	32	**11.18**
10%的均匀噪声，传感器	100%	100%	54	**14.14**
高斯噪声，$\sigma^2=0.1$，传感器	100%	100%	164	**44.33**
高斯噪声，$\sigma^2=0.2$，传感器	100%	100%	193	**85.52**

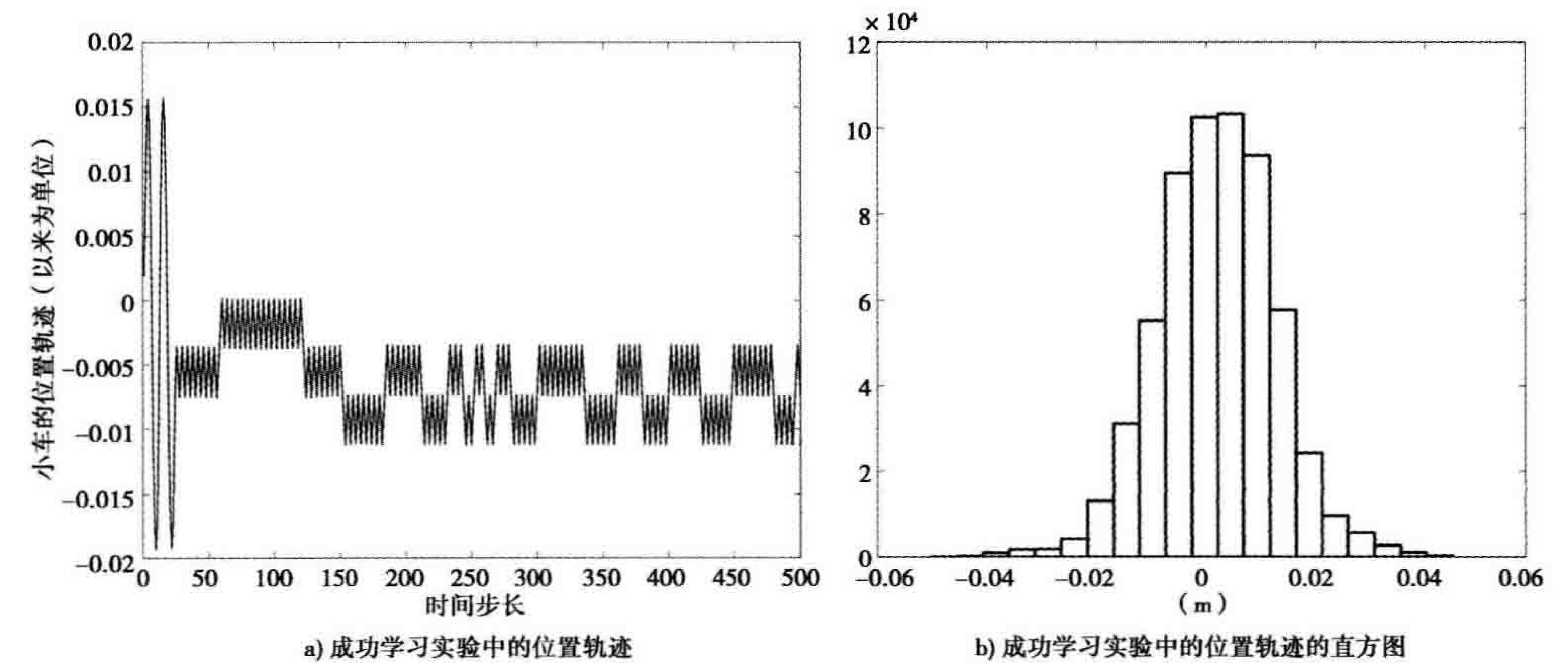


图 5-8 倒立摆的位置轨迹

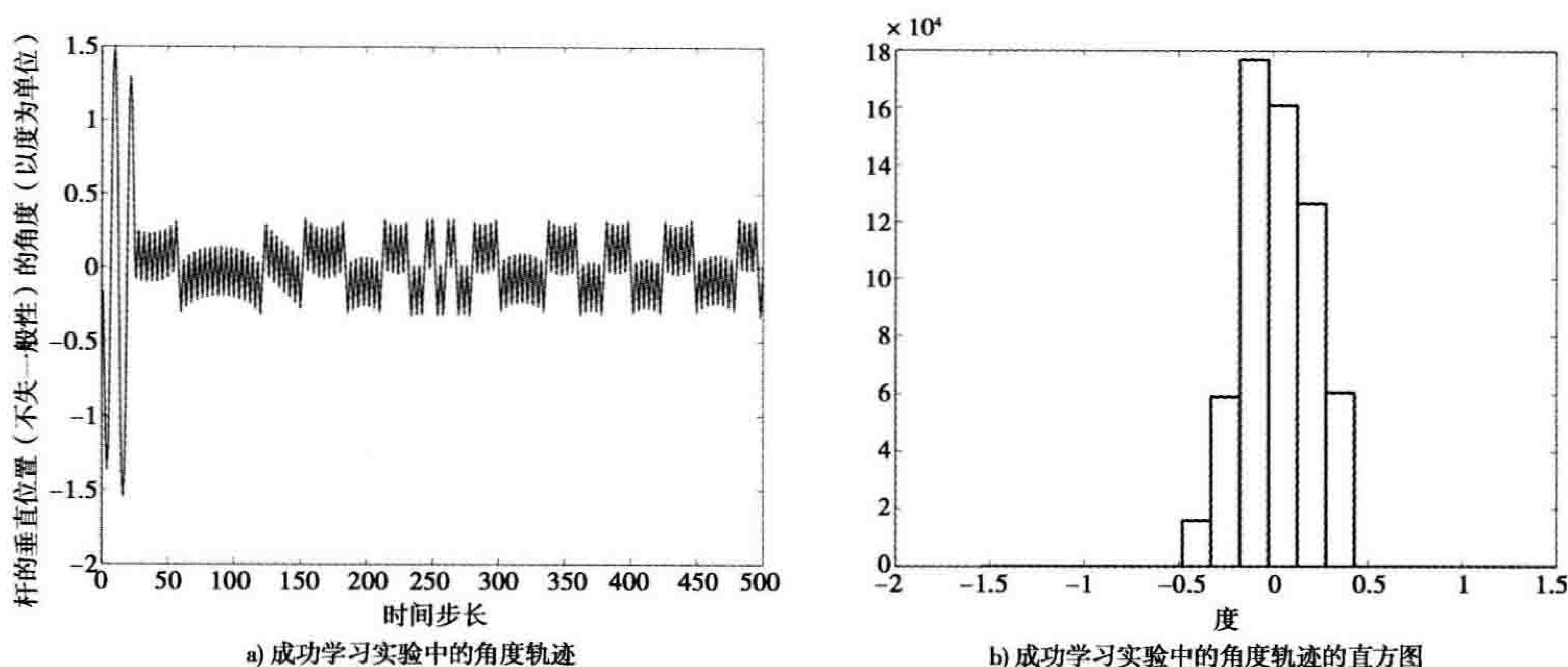


图 5-9 倒立摆的角度轨迹

5.5 总结

- ADP 对通用的类脑智能开发起着至关重要的作用，因为在一般情况下，它是一种接近行动最优策略的通用学习方法。因此，ADP 研究已经在理论方面和实际应用中引起了学术界的极大关注。
- 对于机器智能发展，优化和预测是目标导向行为的两个关键目标，而 ADP 是实现这两个目标的重要方法。就机器智能研究而言，最优化以 Bellman 方程为基础，可定义为随着时间学习做出更好的选择，从而使效用函数最大化，最终实现系统目标。在 ADP 设计中，预测可以被认为是以一种更一般的方法包含了更多重要信息，如来自观察数据的感觉输入以及对未观察到的状态变量的建模和重构。所有的这些，为面向目标行为的智能化系统设计提供了重要信息。
- 本章提出了一种具有多目标表示的分层学习 ADP 构架，该构架可以把优化和预测有效地整合在一起，以实现通用学习。这种架构的核心思想是使用参考网络建立不同层次的内部强化信号表示，以促进学习。参考网络的使用类似于第一代和第二代 ADP 设计中所讨论的“模型网络”。
- ADP 架构通过参考网络预测内部强化信号，为评价网络提供信息丰富的目标表示，从而改进它的优化性能，实现预测。从心理学的角度来看，内部强化信号也可以认为是面向目标行为中“目标”的不同层次的表示。
- 在所提出的 ADP 构架中，学习和适应建立在链式反向传播规则之上，可以实现两种类型的学习策略：序贯学习和协同学习。

参考文献

Al-Tamimi, A., Abu-Khalaf, M., & Lewis, F. L. (2007). Adaptive critic designs for discrete-time zero-sum games with application to h-infinity control. *IEEE Trans. on Syst. Man, Cybern., Part B*, *37*(1), 240–247.

Anderson, C. (1987). Strategy learning with multi-layer connectionist representation. *Proc. International Workshop on Machine Learning*, pp. 103–114.

Anderson, C. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Syst. Mag.*, *9*(3), 31–37.

Balakrishnan, S. N., Ding, J., & Lewis, F. L. (2008). Issues on stability of adp feedback controllers for dynamical systems. *IEEE Trans. on Syst. Man, Cybern., Part B, special issue on ADP/RL, invited survey paper*, *38*(4), 913–917.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Trans. on Syst. Man, Cybern.*, *13*, 834–847.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific.

Derbyshire, S. W. G., Jones, A. K. P., Gyulai, F., Clark, S., Townsend, D., & Firestone, L. L. (1997). Pain processing during three levels of noxious stimulation produces differential patterns of central activity. *Pain*, *73*(3), 431–445.

Enns, R., & Si, J. (2004). Handbook of learning and approximate dynamic programming. In J. Si, A. G. Barto, W. B. Powell, & D. C. Wunsch (Eds.), *Handbook of learning and approximate dynamic programming* (pp. 535–559). Piscataway, NJ: IEEE Press.

Feldkamp, L., Prokhorov, D., Eagen, C., & Yuan, F. (1998). Nonlinear modeling: Advanced black-box techniques. In J. Suykens & J. Vandewalle (Eds.), (pp. 29–53). Norwell, MA: Kluwer.

Feldkamp, L. A., & Prokhorov, D. V. (2003). Recurrent neural networks for state estimation. In K. Narendra (Ed.), *Proc. Workshop on Adaptive And Learning Systems*. New Haven, CT: Yale University.

Ferrari, S., & Stengel, R. F. (2004). *Handbook of learning and approximate dynamic programming*. Piscataway, NJ: IEEE Press.

Fogel, D. B., Hays, T. J., Han, S. L., & Quon, J. (2004). A self-learning evolutionary chess program. *Proc. of the IEEE*, 92, 1947–1954.

Frijda, N. H. (1987). *The emotions*. Cambridge, UK: Cambridge University Press.

Geramifard, A., Bowling, M., & Sutton, R. S. (2006). Incremental least-squares temporal difference learning. *Proc. Twenty-First National Conf. Artificial Intelligence*, pp. 356–361.

Hsieh, J. C., Tu, C. H., & Chen, F. P. (2001). Activation of the hypothalamus characterizes the acupuncture stimulation at the analgesic point in human: A positron emission tomography study. *Neurosci. Lett.*, *307*, 105–108.

Ilin, R., Kozma, R., & Werbos, P. J. (2008). Beyond feedforward models trained by backpropagation: A practical training tool for a more efficient universal approximator. *IEEE Trans. on Neural Netw.*, *19*(6), 929–937.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Liu, D., & Jin, N. (2009). Adaptive dynamic programming for discrete-time systems with infinite horizon and epsilon-error bound in the performance cost. *Proc. IEEE Int. Conf. Neural Netw.*

Melzack, R. (1990). Phantom limbs and the concept of a neuromatrix. *Trends Neurosci.*, *13*, 88–92.

Moriarty, D. E., & Miikulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, *22*, 11–32.

Nicolelis, M. A., Baccala, L. A., Lin, R. C., & Chapin, J. K.. (1995) Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. *Science*, *268*(5215), 1353–1358.

Nicolelis, M. A. L., Lin, R. C., Woodward, D. J., & Chapin, J. K. (1993). Induction of immediate spatiotemporal changes in thalamic networks by peripheral block of ascending cutaneous information. *Nature*, *361*(6412), 533–536.

Peyron, R., Laurent, B., & Garcia-Larrea, L. (2000). Functional imaging of brain responses to pain. a review and meta-analysis. *Neurophysiol Clin.*, *30*, 263–288.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*. Hoboken, NJ: Wiley-Interscience.

Prokhorov, D. V., & Wunsch, D. C. (1997). Adaptive critic designs. *IEEE Trans. on Neural Netw.*, *8*(5), 997–1007.

Puskorius, G. V., & Feldkamp, L. A. (1994). Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks. *IEEE Trans. on Neural Netw.*, *5*(2), 279–297.

Qiao, W., Venayagamoorthy, G., & Harley, R. (2007). DHP-based wide-area coordinating control of a power system with a large wind farm and multiple FACTS devices. *Proc. IEEE Int. Conf. Neural Netw.*, pp. 2093–2098.

Rafols, E. J., Ring, M. B., Sutton, R. S., & Tanner, B. (2005). Using predictive representations to improve generalization in reinforcement learning. *Proc. Twenty-First National Conf. Artificial Intelligence*, pp. 835–840.

Ray, S., Venayagamoorthy, G. K., Chaudhuri, B., & Majumder, R. (2008). Comparison of adaptive critics and classical approaches based wide area controllers for a power system. *IEEE Trans. on Syst. Man, Cybern., Part B*, *38*(4), 1002–1007.

Si, J., Barto, A. G., Powell, W. B., & Wunsch, D. C. (2004). *Handbook of learning and approximate dynamic programming*. Piscataway, NJ: IEEE Press.

Si, J., & Liu, D. (2004). Handbook of learning and approximate dynamic programming. In J. Si, A. G. Barto, W. B. Powell, & D. C. Wunsch (Eds.), *Handbook of learning and approximate dynamic programming* (pp. 125–151). Piscataway, NJ: IEEE Press.

Si, J., & Wang, Y. T. (2001). On-line learning control by association and reinforcement. *IEEE Trans. on Neural Netw.*, *12*(2), 264–276.

Starzyk, J. A., Liu, Y., & He, H. (2006). Challenges of embodied intelligence. *Proc. Int. Conf. Signals and Electronic Syst.*, pp. 534–541.

Sun, P., & Marko, K. (1998). Optimal learning rate for training time lagged recurrent neural networks with the extended kalman filter algorithm. *Proc. IEEE Int. Conf. Neural Netw.*, *2*, 1287–1292.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Vamvoudakis, K., & Lewis, F. L. (2009). Online actor critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Proc. IEEE Int. Conf. Neural Netw.*

Venayagamoorthy, G. K., & Harley, R. G. (2004). Handbook of learning and approximate dynamic programming. In J. Si, A. G. Barto, W. B. Powell, & D. C. Wunsch (Eds.), *Handbook of learning and approximate dynamic programming* (pp. 479–515). Piscataway, NJ: IEEE Press.

Venayagamoorthy, G. K., Harley, R. G., & Wunsch, D. C. (2003). Dual heuristic programming excitation neurocontrol for generators in a multimachine power system. *IEEE Trans. on Industry Applications*, *39*(2), 382–394.

Wang, F. Y., Zhang, H., & Liu, D. (2009). Adaptive dynamic programming: An introduction. *IEEE Comput. Intel. Mag.*, *4*(2), 39–47.

Werbos, P. J. (1977), *Advanced forecasting methods for global crisis warning and models of intelligence*, General System, Yearbook, Vol. 22, 1977.

Werbos, P. J. (1979). Changes in global policy analysis procedures suggested by new methods of optimization. *Policy Analysis and Information Systems*, *3*(1), 27–52.

Werbos, P. J. (1981). *System modeling and optimization*. New York: Springer.

Werbos, P. J. (1983). Solving and optimizing complex systems: lessons from the EIA long-term energy model. In B. Lev (ed.), *Energy Models and Studies*, North Holland, 1983.

Werbos, P. J. (1987). Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Trans. on Syst. Man, Cybern.*, *17*(1), 7–20.

Werbos, P. J. (1988a). Backpropagation: Past and future. *Proc. IEEE Int. Conf. Neural Netw.*, pp. I-343–I-353.

Werbos, P. J. (1988b). Generalization of backpropagation with application to a recurrent gas market model. *Neural Netw.*, *1*, pp. 339–356.

Werbos, P. J. (1989). Backpropagation and neurocontrol: A review and prospectus. *Proc. IEEE Int. Conf. Neural Netw.*, *1*, 209–216.

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proc. IEEE*, 78, 1550–1560.

Werbos, P. J. (1991). An overview of neural networks for control. *IEEE Control Syst. Mag.*, *11*(1), 40–41.

Werbos, P. J. (1992). In D. A. White & D. A. Sofge (Eds.), *Handbook of intelligent control* (pp. 493–525). New York: Van Nostrand.

Werbos, P. J. (1993). Supervised learning: Can it escape its local minimum? *Proceedings WCNN93*, pp. 358–363.

Werbos, P. J. (1994). *The roots of backpropagation: From ordered derivatives to neural networks and political forecasting*. New York: Wiley-Interscience.

Werbos, P. J. (1995). In M. A. Arbib (Ed.), *Handbook of brain theory and neural networks* (pp. 134–139). Cambridge, MA: MIT Press.

Werbos, P. J. (1998a). Dealing with complexity: *A neural networks approach*. Springer.

Werbos, P. J. (1998b). Multiple models for approximate dynamic programming. In K. Narendra (Ed.), *Proc. Yale Conf. on Learning and Adaptive Systems*. New Haven, CT: Yale University.

Werbos, P. J. (1999). *Encyclopedia of electrical and electronics engineering*. New York: Wiley.

Werbos, P. J. (2002). What do neural nets and quantum theory tell us about mind and reality? In K. Yasue, M. Jibu, & T. D. Senta (Eds.), *No matter, never mind: Proceedings of Toward a Science of Consciousness: Fundamental approaches* (pp. 63–87). Springer.

Werbos, P. J. (2004). *Handbook of learning and approximate dynamic programming*. Piscataway, NJ: IEEE Press.

Werbos, P. J. (2005). Automatic differentiation: Applications, theory and implementations, lecture notes in computational science and engineering. In H. M. Bucker, G. Corliss, P. Hovland, U. Naumann, & B. Norris (Eds.), (Vol. 50, pp. 15–34). Springer.

Werbos, P. J. (2007). Using ADP to understand and replicate brain intelligence: the next level design. *IEEE Int. Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 209–216.

Werbos, P. J. (2008). Stable adaptive control using new critic designs *[Online], Available: http://arxiv.org as adap-org/9810001*.

Werbos, P. J. (2009a). Intelligence in the brain: A theory of how it works and how to build it. *Neural Networks*, 200–212.

Werbos, P. J. (2009b). Putting more brain-like intelligence into the electric power grid: What we need and how to do it. *Proc. IEEE Int. Conf. Neural Netw*.

Werbos, P. J., & Pellionisz, A. (1992). Neurocontrol and neurobiology: New developments and connections. *Proc. IEEE Int. Conf. Neural Netw*., 3, 373–378.

White, D. A., & Sofge, D. A. (1992). *Handbook of intelligent control*. New York: Van Nostrand.

第6章

联想学习

6.1 引言

生物智能系统的记忆组织与数字计算机的大不相同(Hawkins & Blakeslee，2004，2007)。生物记忆的特点是分层结构内的联想和自组织，用于分布式的信息存储、预测和回忆。一般而言，联想学习记忆有两种类型：异联想记忆和自联想记忆。异联想记忆把模式成对关联起来(如文字和图片)，而自联想记忆把模式与自身关联，能从模式的部分回想起存储的模式。人类大脑同时使用异联想记忆和自联想记忆来进行学习、行为规划和预测(Rizzuto & Kahana，2001；Brown，Dalloz & Hulme，1995；Murdock，1997)。人类大脑的记忆形成是自组织和数据驱动的。自组织对分层组织形成的作用不仅反映在人脑中，而且也反映在低等脊椎动物的神经系统中(Malsburg，2003)。本章设计并分析了一种能同时异联想和自联想学习的自组织联想记忆模型(Starzyk，He & Li，2007)，该模型以分层形式将稀疏局部连接、自组织处理单元和概率信息传输组织在一起。

6.2 联想学习机制

在机器智能研究领域，已开展了大量联想学习记忆相关的研究工作。例如，在异联想研究方面，Salih、Smith 和 Liu(2000) 提出了利用反馈神经网络的双向联想记忆方法(BAM)。通过用感知训练算法求解一组线性不等式，以实现 BAM 神经网络的设计。Chang 和 Cho(2003) 提出了一种设计二阶非对称双向联想记忆的自适应局部训练规则。Wang(1999) 研究了多联想神经网络(MANN)，并将其成功用于复杂时空序列的学习和检索。仿真结果显示，所提出的多联想神经网络模型具有快速和准确学习的特点，并能储存和检索大量复杂的空间序列模式。至于自联想记忆方面，Hopfield(1982) 的研究是其中的经典，紧随其后出现了许多研究工作。例如，

Vogel 和 Boos(1997) 提出了一种针对稀疏连接网络的自联想记忆算法，所生成的网络，相对于每个神经元的突触数量，具有庞大的信息存储容量。Vogel 和 Boos 推导出了具有二元 Hebbian 突触的两层射影网络(P-net)的存储容量的下界。据报道，给定 1%的容许偏差以激活伪神经元，每个神经元拥有 1000 个突触的 P-net 可以存储超过 1.5×10^6 个训练向量，其中每个向量具有 20 个活动的神经元。Wu 和 Batalama(2000)提出了一种针对前馈联想记忆的有效学习算法，这种记忆采用胜者为王(WTA)机制，并包含一个双层前馈神经网络。最近，Wang 和 Chen(2005)提出了一种基于经验核映射的增强模糊形态学自联想记忆。本章重点介绍分布式分层神经网络组织中的联想学习规则。

6.2.1 单个处理单元的构造

本章介绍的自组织联想记忆由多层阵列处理单元(PE)组成。图 6-1 给出了单个 PE 的接口模型，它包含两个输入(I_1 和 I_2)与一个输出(O)。所有的输入、输出都是双向的，即允许信号向前传播和向后传播。每个 PE 存储分别对应于输入 I_1 和 I_2 的 4 种不同组合($\{I_1, I_2\}$)=$\{00\}$、$\{01\}$、$\{10\}$、$\{11\}$)的观测概率 p_{00}、p_{01}、p_{10} 和 p_{11}。这些用于联想的观测概率值指定了每个 PE 输入空间的数据分布。

图 6-2 给出了一个观测输入数据点分布的例子。观测概率估计如下：

$$p_{00}=\frac{n_{00}}{n_{tot}},p_{01}=\frac{n_{01}}{n_{tot}},p_{10}=\frac{n_{10}}{n_{tot}},p_{11}=\frac{n_{11}}{n_{tot}} \tag{6-1}$$

其中，n_{00}、n_{01}、n_{10} 和 n_{11} 分别是落在 $I_1<0.5\&I_2<0.5$、$I_1<0.5\&I_2>0.5$、$I_1>0.5\&I_2<0.5$、$I_1>0.5\&I_2>0.5$ 区域内的数据点个数。n_{tot} 是数据点的总个数，定义为 $n_{tot}=n_{00}+n_{01}+n_{10}+n_{11}$。对这些概率的动态估计算法详见 Starzyk 和 Wang(2004)。

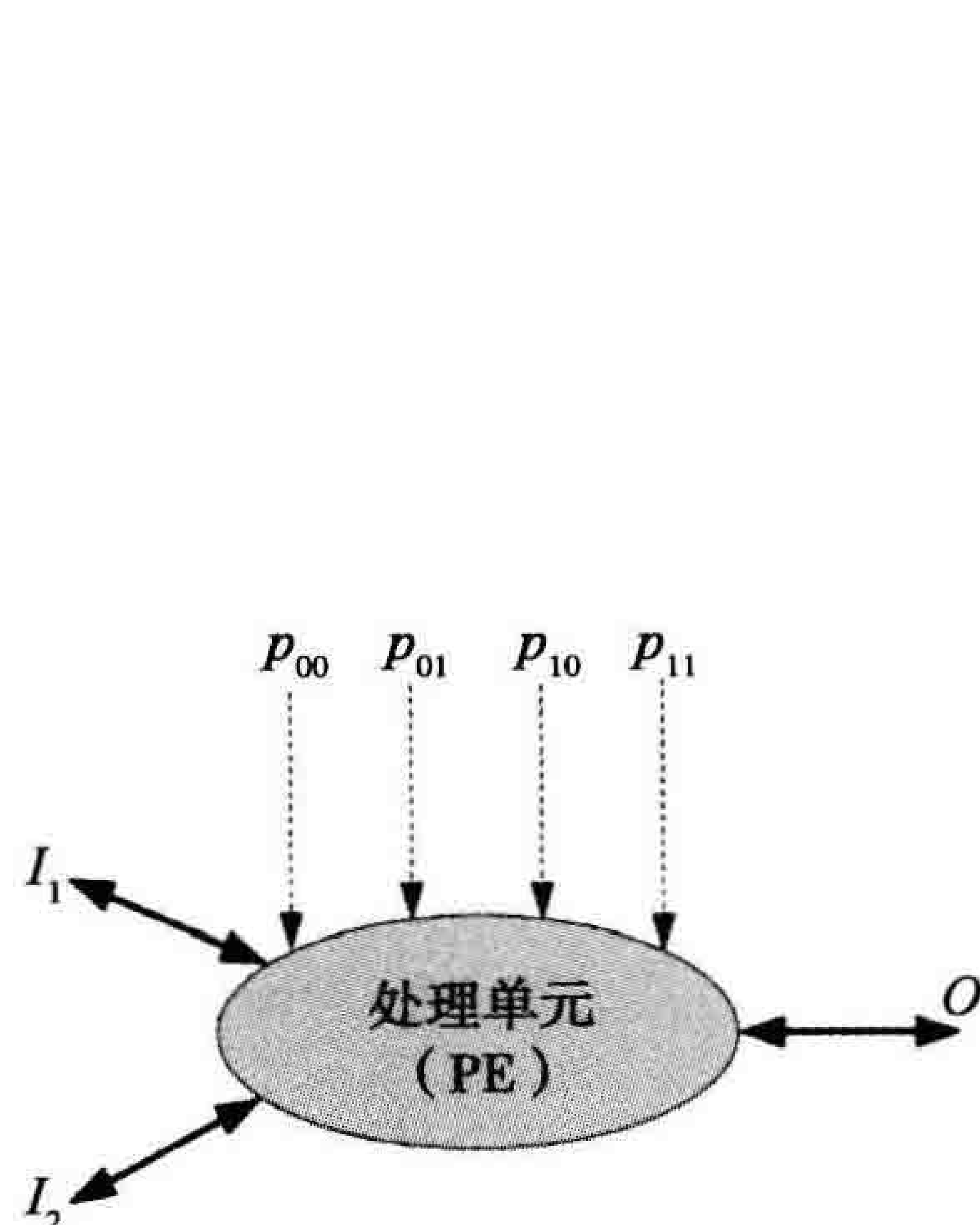


图 6-1 单个 PE 接口模型

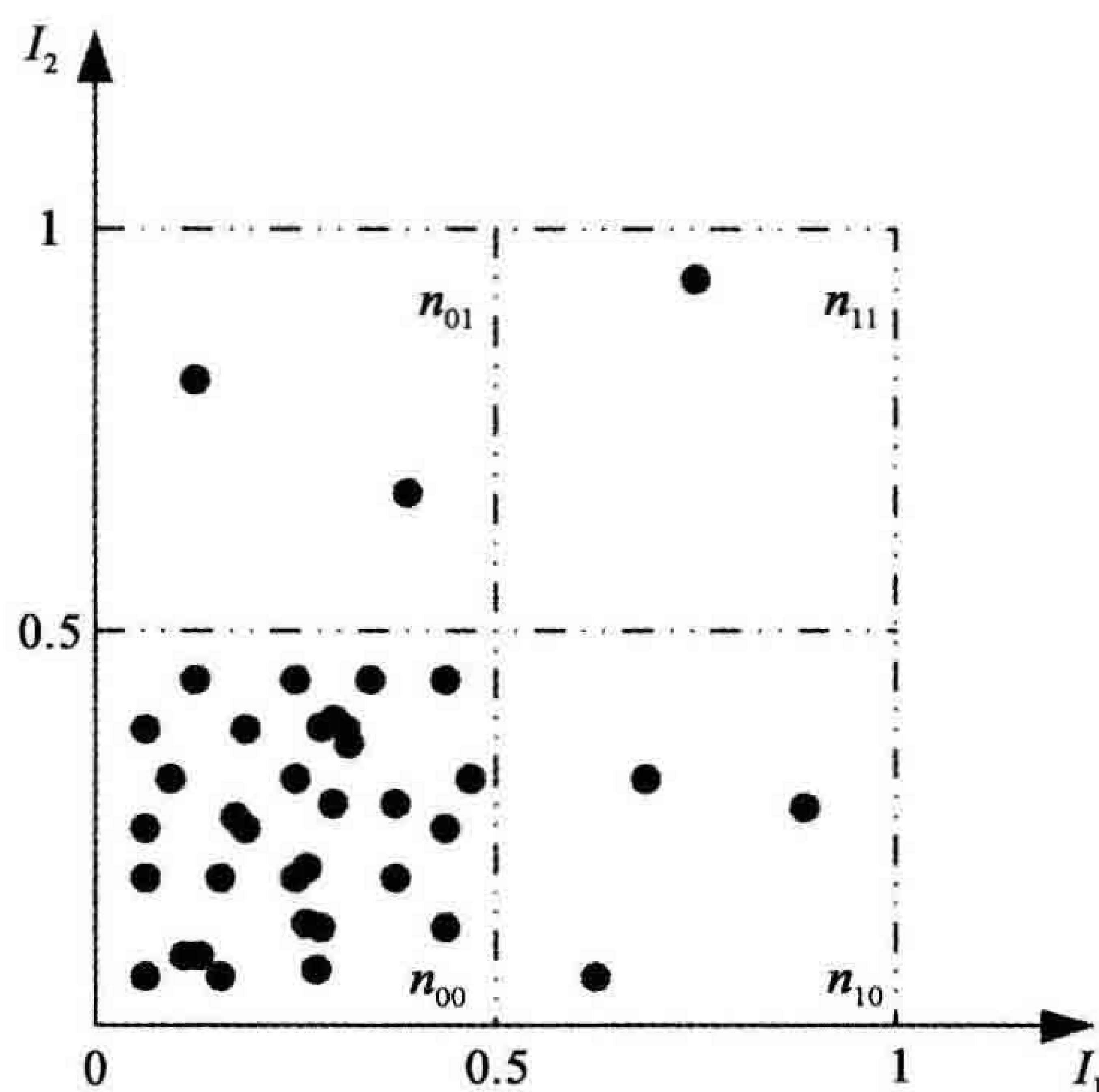


图 6-2 输入空间分布的例子

6.2.2 函数值的自主确定

如图 6-2 所示，每个 PE 根据观测概率 p_{00}、p_{01}、p_{10}和 p_{11}和指定的真值表(表 6-1)来决定其输出函数值 F，输出函数值 f_{00}、f_{01}、f_{10}和 f_{11}由表 6-1 决定(Starzyk 等，2007)：

表 6-1 函数值 F 的自主确定

概率	p_{00}	p_{01}	p_{10}	p_{11}
I_1	0	0	1	1
I_2	0	1	0	1
函数值(F)	f_{00}	f_{01}	f_{10}	f_{11}

根据[算法 6.1]，PE 被激活的概率不超过 0.5，这是由生物神经元稀疏活性决定的(Triesch，2004)。除了生物性动机，低活性也适合有效的能量消耗。表 6-2 给出了两个自主确定函数值 F 的例子。

[算法 6.1] 函数值 F 的自主确定

- 观察输入数据，并根据式(6-1)动态估计 p_{00}、p_{01}、p_{10}和 p_{11}的概率值；
- 寻找与最大概率 p_{ij}(i，j=0，1)对应的输入(I_1，I_2)，并将其对应的输出函数值 F 置为 0；
- 如果最大概率小于 0.5，那么将与最小概率 p_{ij}(i，j=0，1)关联的输入(I_1，I_2)所对应的输出函数值 F 置为 0；
- 如果最大概率与最小概率的和小于 0.5，那么将与第二小概率 p_{ij}(i，j=0，1)关联的输入(I_1，I_2)所对应的输出函数值 F 置为 0；
- 通过上述规则，将对应于输出函数值 F 不为 0 的所有输入组合的 F 值置为 1。

表 6-2 设置 F 值的两个例子

p_{00}	p_{01}	p_{10}	p_{11}	F			
0.4	0.2	0.3	0.1	0	1	1	0
0.4	0.05	0.3	0.25	0	0	1	0

6.2.3 联想学习的信号强度

在联想学习模式下，外部信号在网络中以二元形式出现，而内部信号是 0～1 的半逻辑值，其中 0 和 1 分别是逻辑假值(抑制)和逻辑真值(兴奋)。信号长度定义为

信号值与给定的逻辑阈值之间距离的绝对值(在当前模式中，Th＝0.5。特别地，若信号偏向1或0，可以使用其他阈值)：

$$信号长度(SS) = |信号值 - 逻辑阈值(Th)| \tag{6-2}$$

SS的范围为[0，0.5]。如果SS＝0.5，则信号要么是逻辑真值(兴奋)，要么是逻辑假值(抑制)，分别对应于信号值1或0。如果信号值等于Th，则是未知的(未激活的)，并且SS＝0。当0＜SS＜0.5时，信号处于中级水平。对于一个中级信号，若其值小于Th，则信号电位偏低；若其值大于Th，则信号电位偏高。

图6-3说明了信号长度的定义。在这种方式下，SS提供了一种确定何时触发联想的连贯机制，并且当多于一个反馈信号值出现在PE的输出端口(见“反馈操作”小节)时，有助于决定反馈信号值。

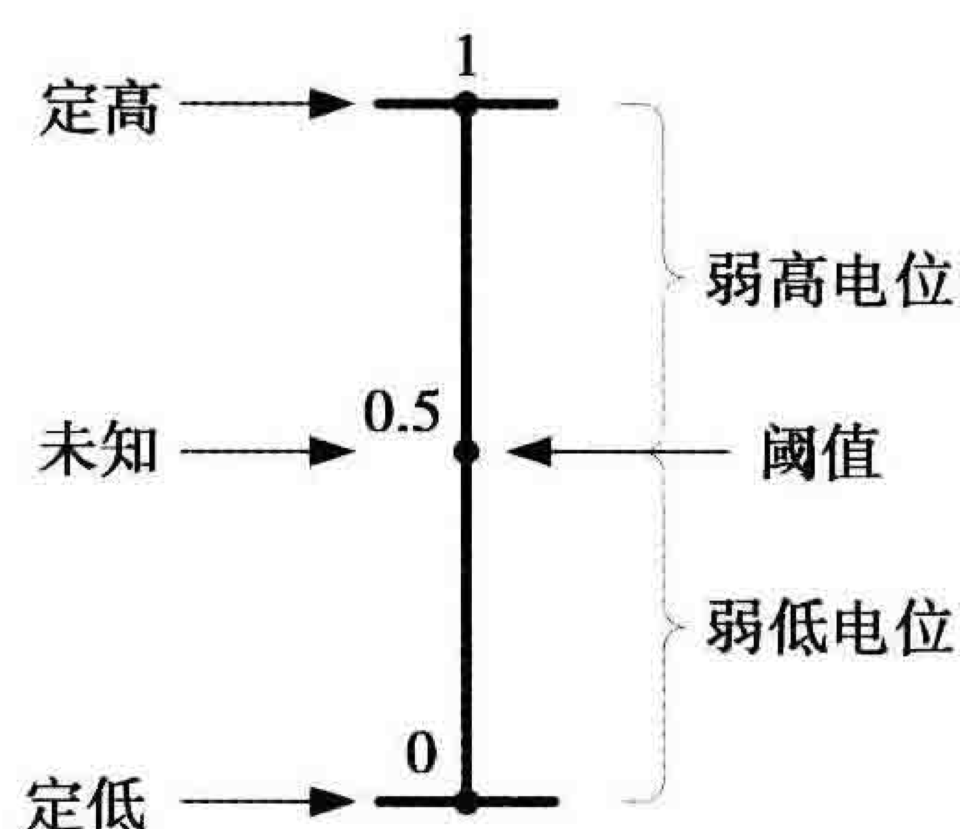


图6-3 信号长度及其半逻辑值

6.2.4 联想学习原则

在训练阶段，每个PE计算它在n_{00}、n_{01}、n_{10}和n_{11}中的输入数据点，并估计对应概率p_{00}、p_{01}、p_{10}和p_{11}。每个PE在训练阶段的目的是寻找各自输入之间的潜在联系，这种联系由对应的概率所记忆，并且在测试阶段用来进行联想。

图6-4说明了在测试阶段用于推断未知信号值的3种联想类型。

1)只输入联想(IOA)。在测试阶段，如果一个输入已定义，而另一个输入和来自其他PE所接收的输出反馈信号O_f未定义(例如，若$I_1=0$，$I_2=0.5$且$O_f=0.5$，如图6-4a所示)，则这个PE通过与I_1的关联来确定I_2。

2)只输出联想(OOA)。如果输入I_1和I_2均未定义，则这两个输入由反馈信号O_f确定(见图6-4b)。例如，考虑图6-2中的例子，在给定的PE中，大部分输入数据点分布在左下角($I_1<0.5\&I_2<0.5$)。假设相关概率为$p_{00}=0.8$，$p_{01}=0.07$，$p_{10}=0.1$和$p_{11}=0.03$，根据[算法6.1]，$F=\{0，1，1，1\}$。在这种情况下，若

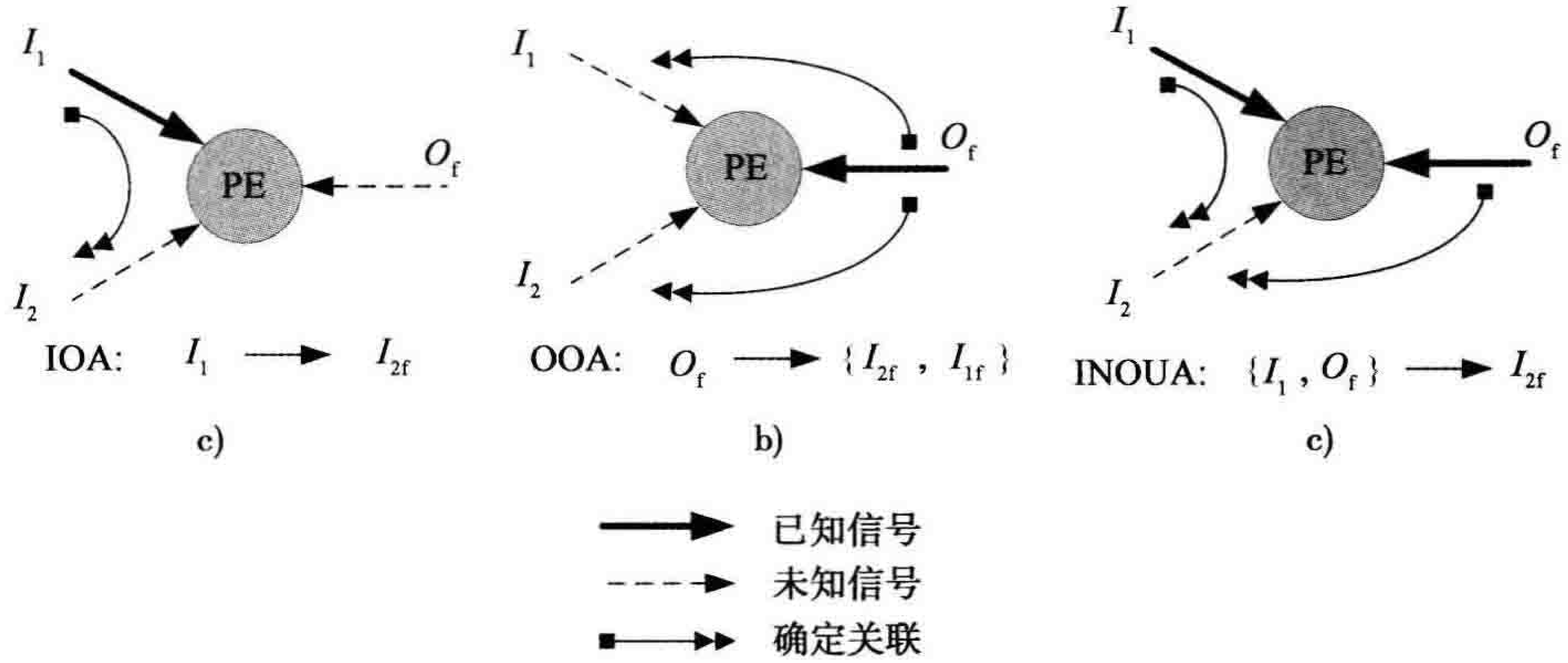


图 6-4 PE 的 3 种联想类型

$O_f=0$ 且 $I_1=I_2=0.5$，则该 PE 将推断出输入 I_1 和 I_2 均为 0(在这里，I_{1f}和 I_{2f}用来表示 I_1 和 I_2 的反馈信号，以区别 I_1 和 I_2 对应的前馈信号)。另一方面，若接收的输出反馈信号 $O_f=1$，输入反馈值 I_{1f}和 I_{2f}处于中间水平，则可通过数据分布概率估计其值。

3)输入-输出联想(INOUA)。若一个输入信号和输出反馈信号 O_f 已定义，另一个输入信号未定义，则 PE 将通过观测到的概率推断出另一个输入信号，如图 6-4c 所示。

现在用方程式来表示联想学习机制的数学基础(Starzyk 等，2007)，共有 4 种情形：

情形 1：给定两个输入 $V(I_1)$和 $V(I_2)$的半逻辑值，决定其输出值 $V(O)$。

假设一个 PE 接收输入值 $V(I_1)=m$，$V(I_2)=n$，则

$$V(O)=\frac{p(I_1=1,I_2=1,F=1)}{p(I_1=1,I_2=1)}\cdot V_{11}+\frac{p(I_1=0,I_2=1,F=1)}{p(I_1=0,I_2=1)}\cdot V_{01}+\frac{p(I_1=1,I_2=0,F=1)}{p(I_1=1,I_2=0)}\cdot V_{10}+\frac{p(I_1=0,I_2=0,F=1)}{p(I_1=0,I_2=0)}\cdot V_{00} \tag{6-3}$$

其中，V_{11}、V_{01}、V_{10}和 V_{00}的定义如下：

$$\begin{aligned} V_{11}&=mn\\ V_{01}&=(1-m)n\\ V_{10}&=m(1-n)\\ V_{00}&=(1-m)(1-n) \end{aligned} \tag{6-4}$$

$p(I_1=1,I_2=1,F=1)$、$p(I_1=1,I_2=1)$等是联合概率，可以从表 6-1 中利用概率 p_{00}、p_{01}、p_{10}和 p_{11}获得。例如，如果某个 PE 具有 $F=\{0,1,1,1\}$，则

$$p(I_1=1,I_2=1,F=1)=p_{11}$$
$$p(I_1=0,I_2=0,F=1)=0 \tag{6-5}$$
$$p(I_1=0,I_2=1)=p_{01}$$

在训练和测试阶段，当信号正向传播时，这种情形是必需的。在当前模型中，输入和输出的半逻辑值分别为0、1和0.5。若使用其他半逻辑值，这些方程式仍然成立。

情形2：给定一个输入$V(I_1)$或$V(I_2)$的半逻辑值和一个未定义的输出$V(O)$，确定另一个输入值。

这种情形对应于IOA，如图6-4a所示。给定$V(I_1)$，确定未定义的$V(I_2)$的情形如下：

$$V(I_2)=\frac{p(I_1=1,I_2=1)}{p(I_1=1)}\cdot V(I_1)+\frac{p(I_1=0,I_2=1)}{p(I_1=0)}\cdot(1-V(I_1)) \tag{6-6}$$

其中，

$$p(I_1=1)=p_{10}+p_{11}$$
$$p(I_1=0)=p_{00}+p_{01} \tag{6-7}$$

此时$V(I_2)$给定，确定$V(I_1)$，在式(6-6)中I_1和I_2是可交换的。在测试阶段，当信号反向传播时，这种情形是必需的。

情形3：给定输出值$V(O)$，确定输入值$V(I_1)$和$V(I_2)$。

$$V(I_1)=\frac{p(F=1,I_1=1)}{p(F=1)}\cdot V(O)+\frac{p(F=0,I_1=1)}{p(F=0)}\cdot(1-V(O))$$
$$V(I_2)=\frac{p(F=1,I_2=1)}{p(F=1)}\cdot V(O)+\frac{p(F=0,I_2=1)}{p(F=0)}\cdot(1-V(O)) \tag{6-8}$$

这种情形对应于OOA，如图6-4b所示。$p(F=1)$和$p(F=0)$是根据表6-1中F值的概率决定的。例如，若$F=\{0\ 1\ 0\ 1\}$，则$p(F=1)=p_{01}+p_{11}$，$p(F=0)=p_{00}+p_{10}$。当信号反向传播时，这种情况是必需的。

情形4：给定一个输入值$V(I_1)$或$V(I_2)$，以及输出值$V(O)$，确定另一个输入值$V(I_2)$或$V(I_1)$。

这种情形对应于6-4c中的INOUA。例如，给定$V(I_1)$和$V(O)$，确定$V(I_2)$的情形如下：

$$V(I_2)=\frac{p(I_1=1,F=1,I_2=1)}{p(I_1=1,F=1)}\cdot \hat{V}_{11}+\frac{p(I_1=0,F=1,I_2=1)}{p(I_1=0,F=1)}\cdot \hat{V}_{01}$$

$$+\frac{p(I_1=1,F=1,I_2=1)}{p(I_1=1,F=0)}\cdot\hat{V}_{10}+\frac{p(I_1=0,F=0,I_2=1)}{p(I_1=0,F=0)}\cdot\hat{V}_{00} \tag{6-9}$$

其中，$\hat{V}_{11}$、$\hat{V}_{10}$、$\hat{V}_{01}$和$\hat{V}_{00}$的确定方法如下：

$$\hat{V}_{11}=\begin{cases}V(I_1)\cdot V(O) & \begin{cases}X & X & 0 & 1\\ X & X & 1 & 0\end{cases}\\ 0 & \quad X \quad X \quad 0 \quad 0\\ V(I_1) & \quad X \quad X \quad 1 \quad 1\end{cases} \tag{6-10}$$

$$\hat{V}_{10}=\begin{cases}V(I_1)\cdot(1-V(O)) & \begin{cases}X & X & 0 & 1\\ X & X & 1 & 0\end{cases}\\ V(I_1) & \quad X \quad X \quad 0 \quad 0\\ 0 & \quad X \quad X \quad 1 \quad 1\end{cases} \tag{6-11}$$

$$\hat{V}_{01}=\begin{cases}(1-V(I_1))\cdot V(O) & \begin{cases}0 & 1 & X & X\\ 1 & 0 & X & X\end{cases}\\ 0 & \quad 0 \quad 0 \quad X \quad X\\ 1-V(I_1) & \quad 1 \quad 1 \quad X \quad X\end{cases} \tag{6-12}$$

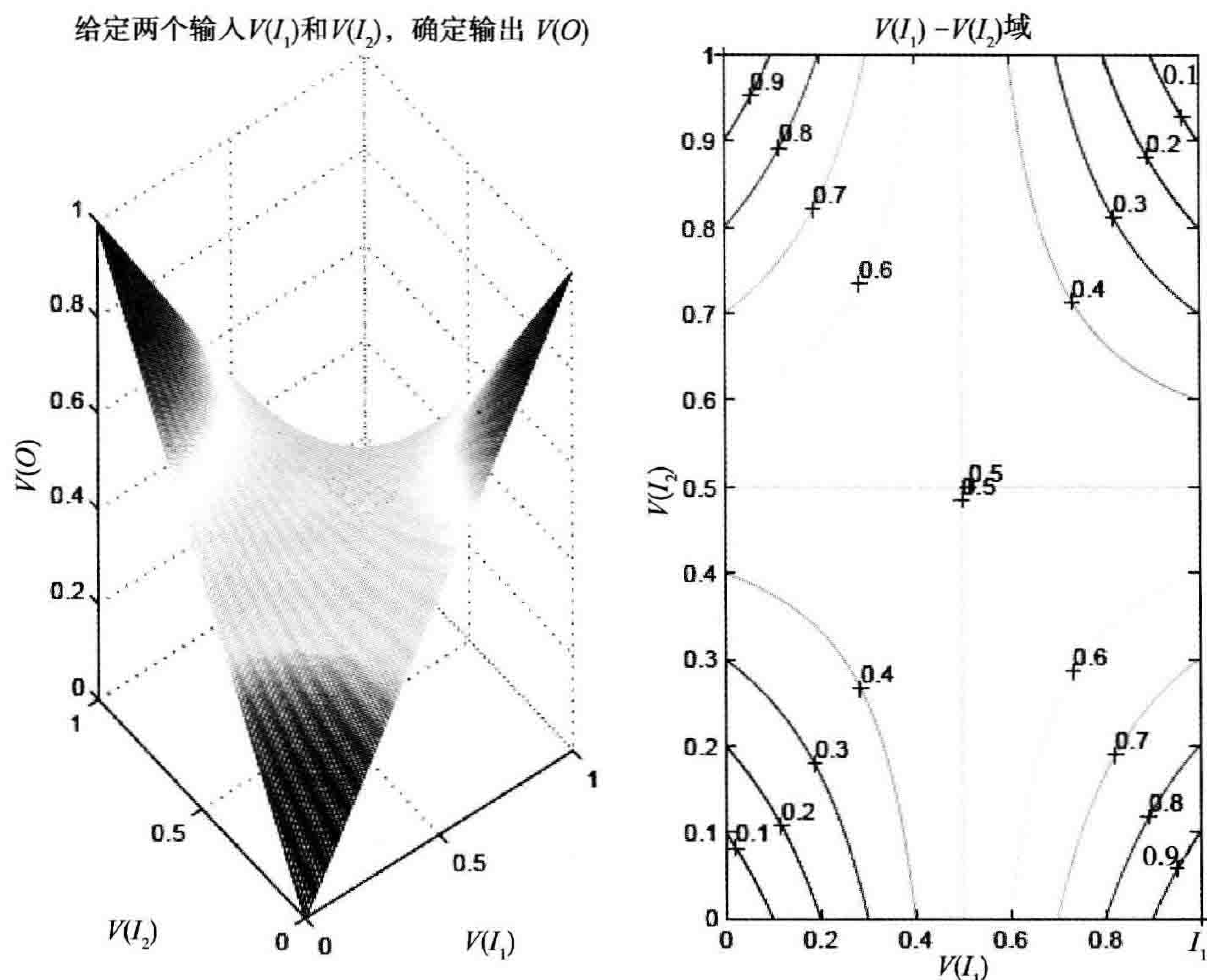
$$\hat{V}_{00}=\begin{cases}(1-V(I_1))\cdot(1-V(O)) & \begin{cases}0 & 1 & X & X\\ 1 & 0 & X & X\end{cases}\\ 1-V(I_1) & \quad 0 \quad 0 \quad X \quad X\\ 0 & \quad 1 \quad 1 \quad X \quad X\end{cases} \tag{6-13}$$

式(6-10)～式(6-13)的条件与表 6-1 中的 F 值有关，其中的“X”是不考虑的值，意味着其值可以是 0 或 1。例如，若一个 PE 接收到 $V(I_1)=m$ 和 $V(O)=t$，该 PE 的函数值是 $F=\{0 \quad 1 \quad 1 \quad 1\}$，那么将得出下列结果：

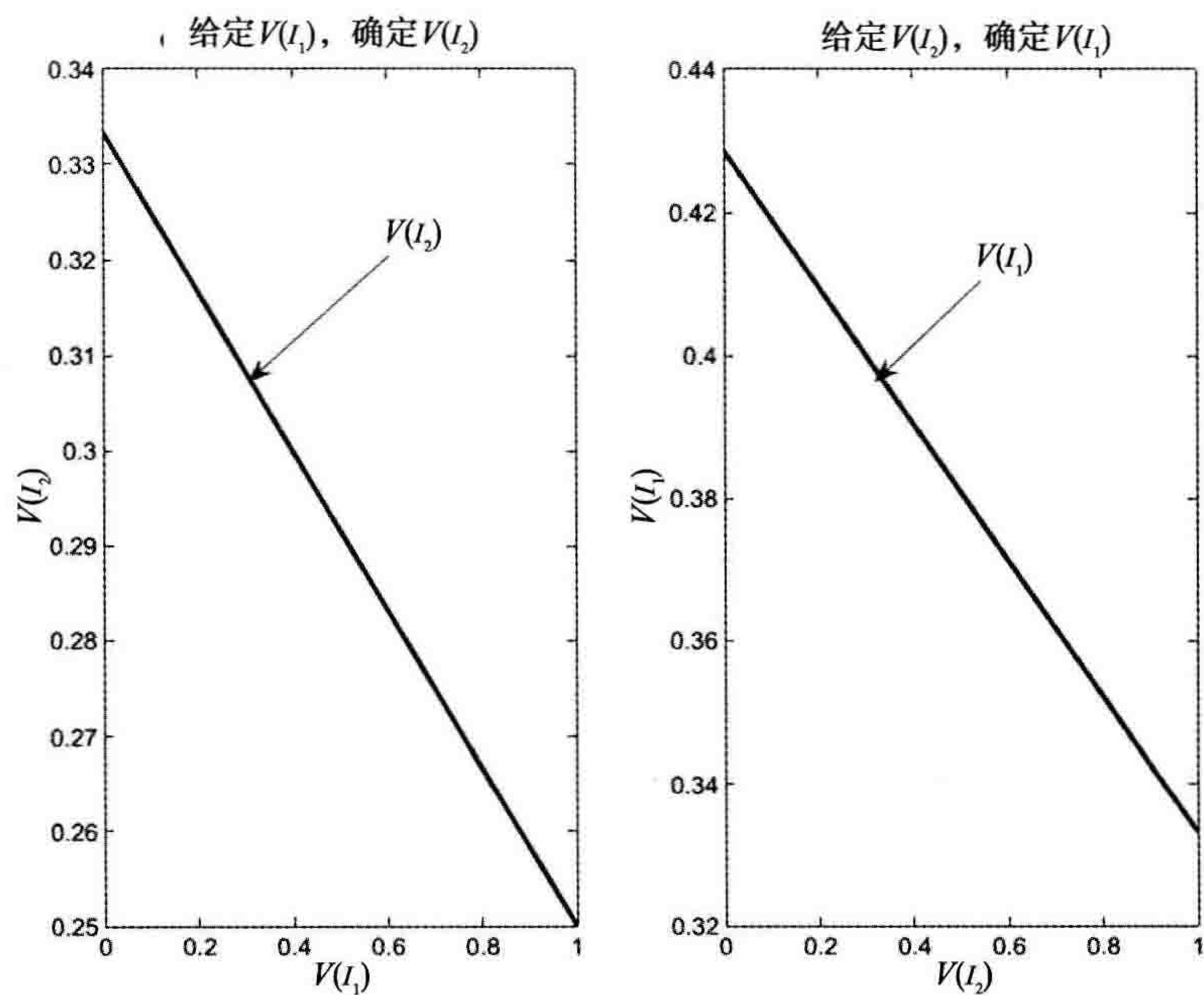
$$\begin{aligned}\hat{V}_{11}&=m\\ \hat{V}_{10}&=0\\ \hat{V}_{01}&=(1-m)\times t\\ \hat{V}_{00}&=(1-m)\times(1-t)\end{aligned} \tag{6-14}$$

当给定 $V(I_2)$和 $V(O)$时，则只需在式(6-9)～式(6-13)中交换 I_1 和 I_2，就可以确定 $V(I_1)$。在信号反向传播时，此种情况是必需的。

在详细讨论记忆结构和操作过程之前，首先展示关于联想学习的这种概率推断的仿真结果。假设一个 PE 观测到的数据分布概率为 $p_{00}=0.4$、$p_{01}=0.2$、$p_{10}=0.3$ 和 $p_{11}=0.1$，根据[算法 6.1]，该 PE 将自动确定其函数值 F：$F=\{0\ 1\ 1\ 0\}$。利用上述 4 种情形所描述的概率推断方法，图 6-5 说明了对应的关联信号信息。

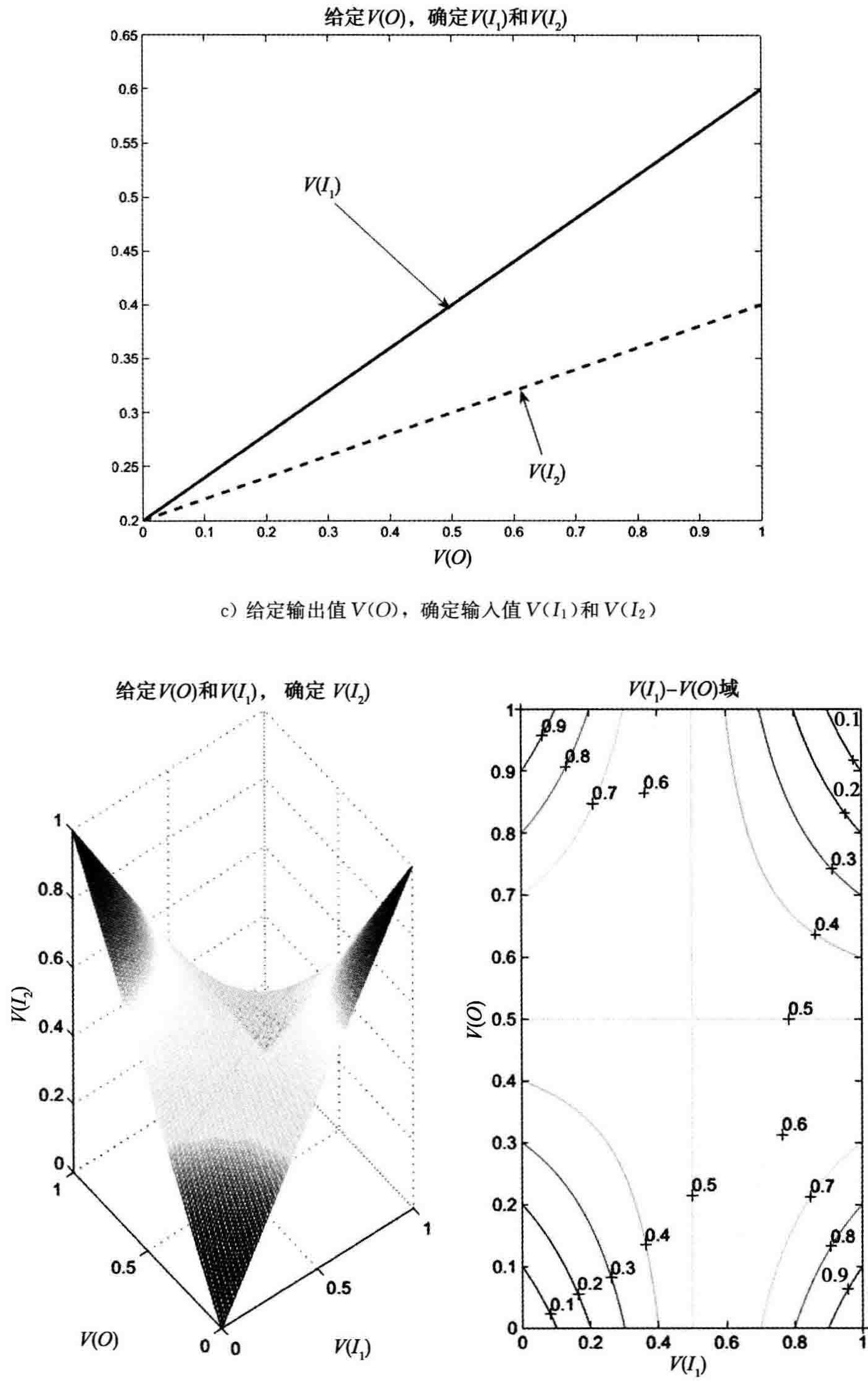


a) 给定两个输入 $V(I_1)$和 $V(I_2)$，确定输出 $V(O)$



b)给定一个输入 $V(I_1)$或 $V(I_2)$、一个未定义的输出 $V(O)$，确定另一个输入的值

图 6-5　4 种信息联想情形的仿真结果



c) 给定输出值 $V(O)$，确定输入值 $V(I_1)$和 $V(I_2)$

d) 给定一个输入值 $V(I_1)$或 $V(I_2)$、输出值 $V(O)$，确定另一个输入值 $V(I_2)$或 $V(I_1)$

图 6-5 （续）

从图 6-5a 中可以看出，该 PE 执行了一个广义异或函数。为了验证这些仿真结果，以图 6-5b 为例。基于观测到的概率分布 $p_{00}=0.4$、$p_{01}=0.2$、$p_{10}=0.3$ 和 $p_{11}=0.1$，可以看出，若输入 I_1 很小，则另一个输入 I_2 很可能也很小(由于 $p_{00}>p_{11}$)。图 6-5b 证实了这一仿真结果。利用类似的方法，可以检测其他所有的仿真结果与数据分布概率是一致的。因此，利用概率推断机制，可以发现每个 PE 接收到的输入数据之间的关系，并在测试阶段做出正确的联想，以恢复丢失的数据信息。

6.3 分层神经网络中的联想学习

6.3.1 网络结构

整个记忆网络是稀疏连接的自组织处理单元的分层结构。阵列中的每个 PE 可根据对输入数据的响应来动态调整它的函数，从而进行自组织。记忆网络中的所有 PE 都有两个输入 I_1 和 I_2，一个输出 O。对于 I_1、I_2 和输出，每个端口都有一个与之相关的反馈信号，分别记为 I_{1f}、I_{2f}和 O_f。所有的 PE 都是相同的，其功能取决于各自的概率分布。在本节所介绍的分层结构中，每个 PE 只与其下层或上层的 PE 连接。这样的分层连接适合于硬件实现及时间控制，并与目标表示的复杂性相关。进一步，PE 来自于感觉输入、更抽象多变的目标表示或特征。每个 PE 更可能与其他 PE 在较短的欧氏距离内连接，尽管少数 PE 可以远距连接。这种组织结构可以在神经元趋于局部连接的生物记忆中观测到。因此，横向连接概率是高斯分布和均匀分布的叠加。

6.3.2 网络操作

1. 前馈操作

在训练和测试阶段，前馈操作不可或缺。图 6-6 展示了一个联想记忆的前馈网络结构。为了简单表述，这里只列举了 4 层，其中每层具有 6 个 PE。从 PE1 到 PE11、PE18 到 PE21 的粗线表示两个远程连接的例子。

在训练阶段，所有的外部传感器的数据输入到网络中。每个 PE 根据输入信息估计对应的概率 $p_{ij}(i,j=0,1)$，并利用 6.2.4 节所描述的情形 1 决定其输出函数。在测试阶段，某些输入值是未定义的(信号值置为 0.5)。当输入信号未定义时，该 PE 的输出信号也不能确定。否则，将利用训练阶段建立的概率来决定，正如 6.2.4

节的情形 1 所示。事实上，训练和测试之间的区别是人为设定的，这是由于网络一直在学习，一直在更新所有接收确定输入的 PE 的输入概率。

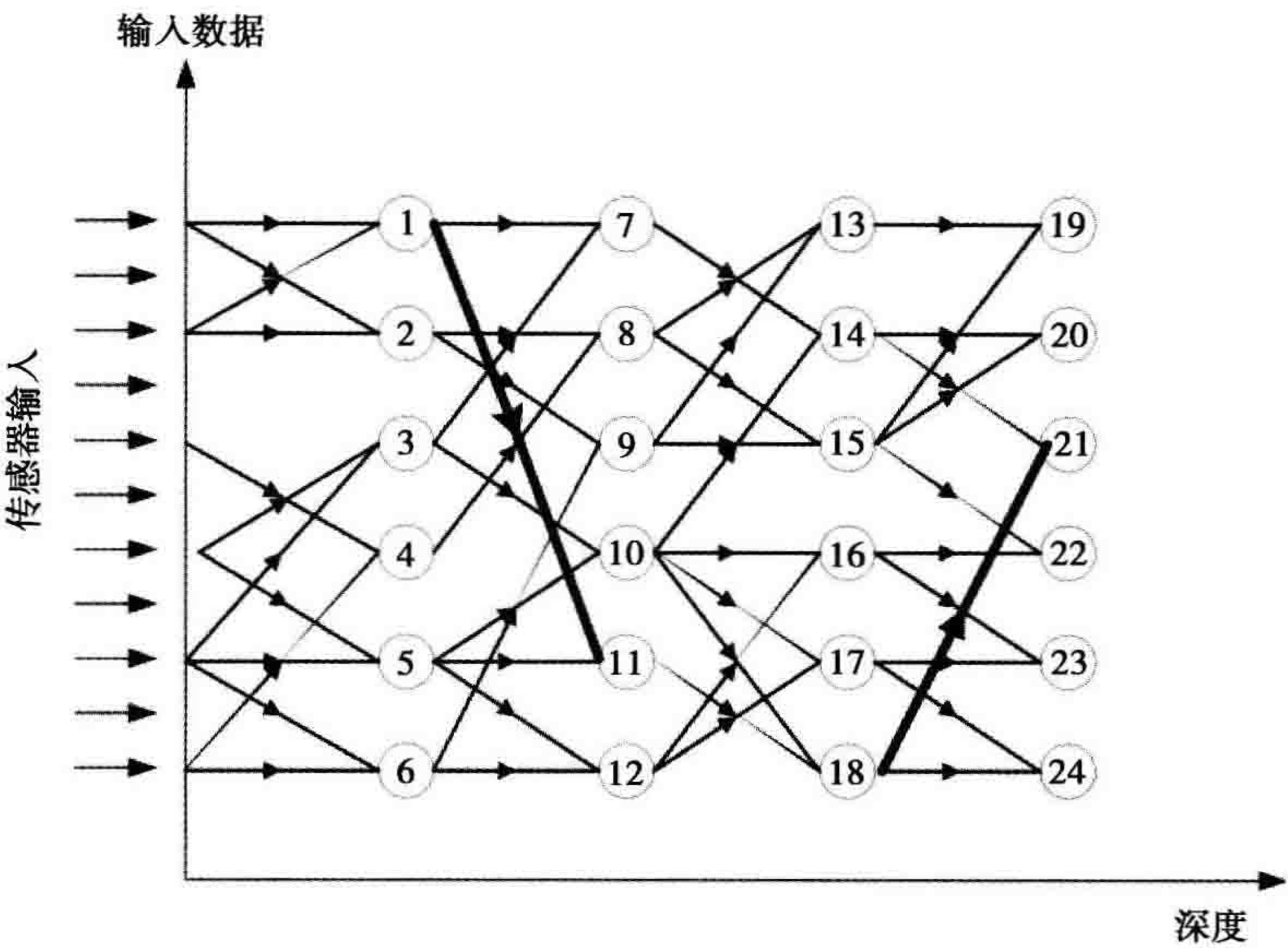


图 6-6　前馈操作网络的例子

2. 反馈操作

反馈操作对于网络建立正确的关联、恢复输入数据的缺失部分必不可少(Starzyk 等，2007)。图 6-7 展示了测试阶段的反馈结构。网络连接与图 6-6 所示的相同，为了清楚说明反馈信号，一些反馈条件没有在图中显示出来。在测试阶段，有部分信息是不确定的，就像分类应用(所有的类标签未知，联想记忆网络中只有输入值的特征)或图像恢复应用(部分图像可能被分块或是未定义的)。在这两种情况下，网络将利用联想来确定未知的信号值。

在图 6-7 中，阴影所示的 PE 是相关联的，下面将说明如何利用联想来恢复未知的值，这里使用了 6.2.4 节所描述的 3 种联想类型。如图 6-7 所示，本节所描述的模型可以根据输入数据的复杂性自己确定反馈结构中所使用的联想的深度。在许多复杂应用中，这种自组织能力有很强的灵活性。

为了说明反馈联想机制，现讨论如图 6-8 所示的部分联想记忆。

(1)当 $T=k$ 时

PE1 有两个确定的输入(I_{1_1} 是低电位，I_{2_1} 是高电位，下标表示 PE 的编号)。这种情况下，PE1 根据 6.2.4 节中所描述的概率学习算法(情形 1)确定其输出值。假设得到的输出为 $O_1=0$，PE2 具有两个未知输入。于是，PE2 输出一个未定义的值

$O_2=0.5$。这里未使用联想或反馈值。

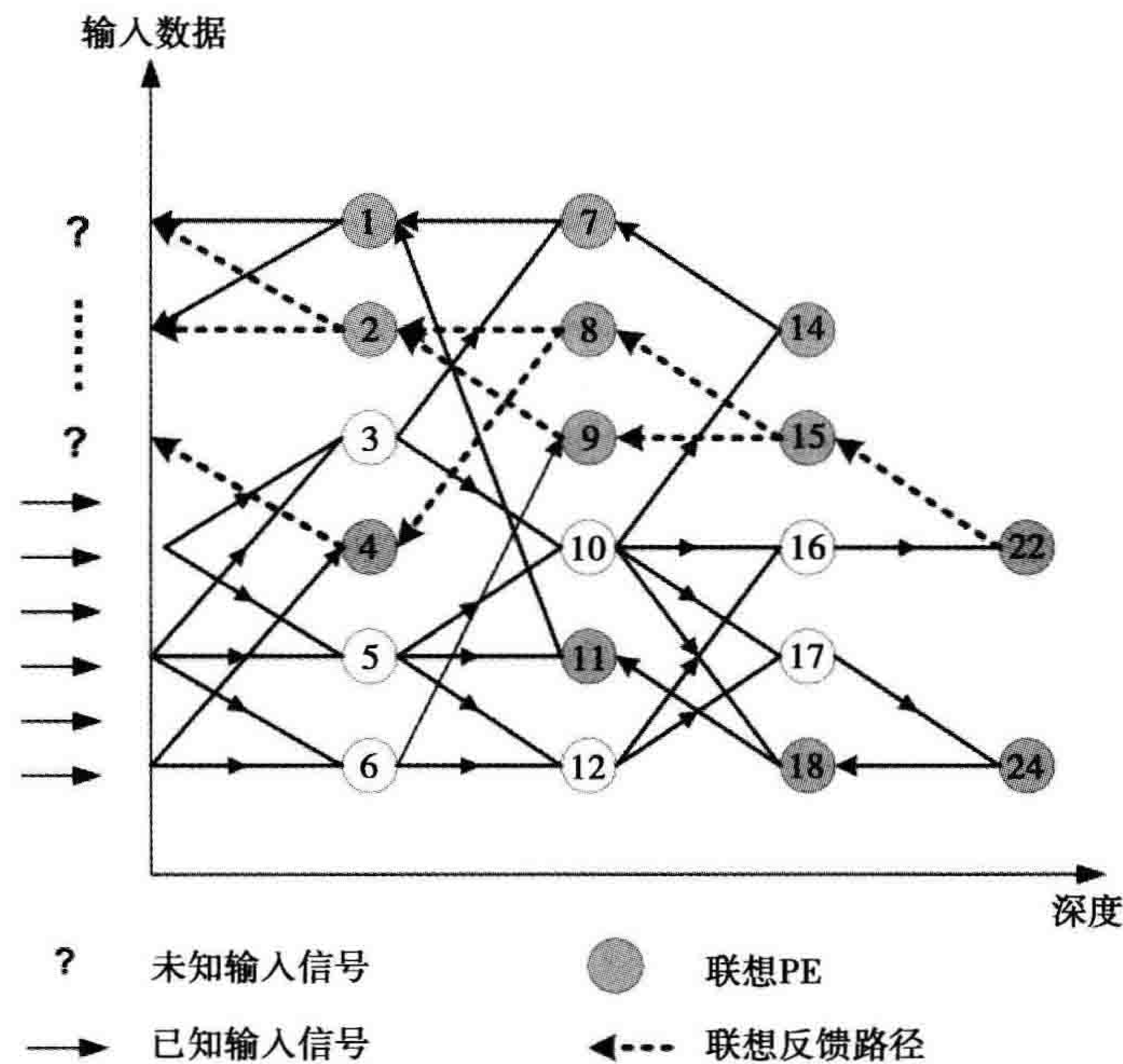


图 6-7 测试阶段的反馈结构示例

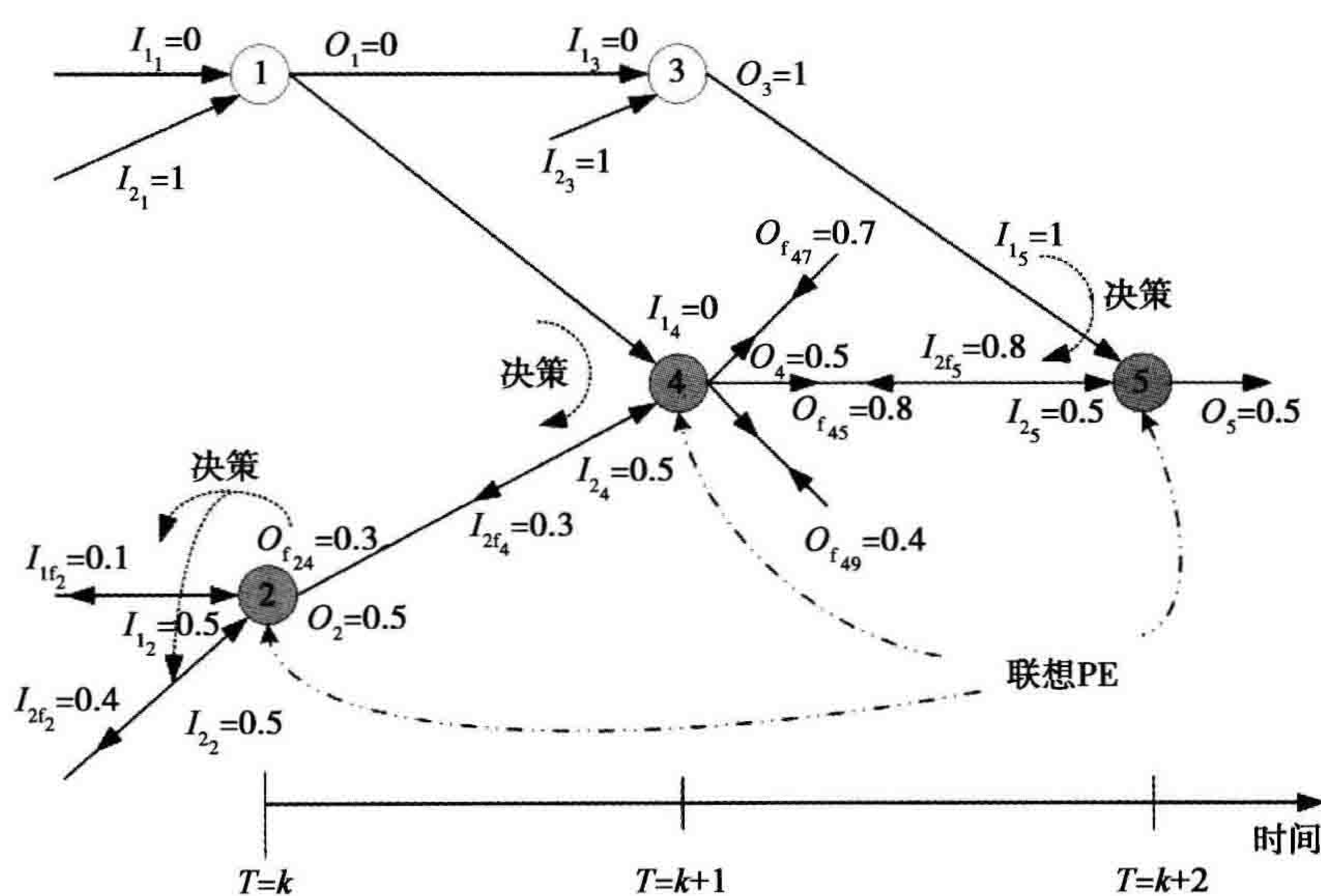


图 6-8 联想记忆的反馈机制

(2)当 $T=k+1$ 时

由于PE3的两个输入都已定义，根据6.2.4节所描述的情形1可以确定其输出值。由于 $I_{2_4}=0.5$ 且PE1的输出已经定义($O_1=0$)，PE4使用IOA确定反馈信号 I_{2f_4}。假

设 I_{2f_4} 的反馈信号是0.3，该反馈信号 I_{2f_4} 将变成PE2的输出反馈信号($O_{f_{24}}=0.3$)。这里的第一和第二下标分别表示反馈目标和反馈源(4到2)。由于PE2的两个输入均未定义，$O_{f_{24}}$ 触发从输出到两个输入(OOA)的联想，这对应于6.2.4节所描述的情形3。对于PE2，假设训练信息导致 $I_{1f_2}=0.1$、$I_{2f_2}=0.4$，在低层，这些输入反馈信号将输出反馈信号变成较低层的PE2的目标，并可能触发其他联想。

(3)当 $T=k+2$ 时

由于 I_{1_5} 已经被定义($I_{1_5}=1$)且 I_{2_5} 未定义，根据6.2.4节所描述的情形2，IOA联想将确定PE5的输入反馈信号(假设 $I_{2f_5}=0.8$)。同时，假设PE4从3个不同的PE(PE7、PE5、PE9)接收3个反馈信号，其中 $O_{f_{47}}=0.7$、$O_{f_{45}}=0.8$、$O_{f_{49}}=0.4$，在这种情形下，选择具有最大强度的信号作为PE4的输出反馈信号，即

$$O_f = \max(\mathrm{SS}(O_{f_{4i}})) \tag{6-15}$$

这里PE4从PEi接收反馈信号，i 为7、5或9。

在这种情况下，有

$$\begin{aligned}\mathrm{SS}(O_{f_{47}}) &= |0.7-0.5| = 0.2\\ \mathrm{SS}(O_{f_{45}}) &= |0.8-0.5| = 0.3\\ \mathrm{SS}(O_{f_{49}}) &= |0.4-0.5| = 0.1\end{aligned} \tag{6-16}$$

因此，PE4的反馈信号为 $O_{f_4}=0.8$。该信号触发PE4的联想。对于PE4，I_{1_4} 已定义($I_{1_4}=0$)，O_{f_4} 也已定义($O_{f_4}=0.8$)，I_{2_4} 未定义，因此，反馈信号 I_{2f_4} 是基于IOA的。根据6.2.4节所描述的情形4，假设反馈信号 $I_{2f_4}=0.2$，但在上一步，信号 I_{2f_4} 为0.3。因为在特定的系统中，一个信号只能有一个值，因此需要解决两个信号值的问题。这是通过选择具有最大信号强度的信号来实现的。在此例中，0.2(SS=0.3)比0.3(SS=0.2)强，所以 I_{2f_4} 更新为0.2。此时，更新的 I_{2f_4} 成为PE2的输出反馈信号，根据6.2.4节所描述的情形3，它反过来触发PE2的联想。

总之，所提出的记忆网络通过建立必要的联想来追踪上一层的信号，从而最终决定未知的信号值。应当注意的是，更新的输入反馈信号(I_{1f}、I_{2f})和输出反馈信号(O_f)不能前向传播到更高层，因此在网络中，这可能引起不稳定和振荡。

6.4 实例研究

在本章，我们用来自UCI机器学习库(Asuncion & Newman，2007)的Iris数据库和两个图像恢复问题来说明自组织记忆模型中的异联想和自联想的应用。

6.4.1 异联想应用

Iris 数据库(Asuncion & Newman，2007)可以用来测试记忆模型的分类性能。该数据库有 3 个类(Iris Setosa、Iris Versicolour 和 Iris Virginica)和 4 个数字特征(sepal length、sepal width、petal length 和 petal width)。

在这个实例研究中，我们用 N 位滑动条编码机制来编码输入的数据(Starzyk 等，2007；Starzyk，Zhu，& Li，2006)。假设最大值和最小值分别编码为 V_{max} 和 V_{min}，滑动条的长度定义为 $N-L=V_{max}-V_{min}$，特征值编码为 V。在编码输入中，从$(V-V_{min})+1$ 到$(V-V_{min})+L$ 的位被记为 1，而剩余的位记为 0，如图 6-9 所示。

如图 6-10 所示，类标签用相似的方法编码，从$(C_i-1)\times M$ 到 $C_i\times M$ 的 M 位编码为 1，而剩余的 $M\times 2$ 位编码为 0。这里，$C_i=1$，2 和 3 表示 3 类 Iris 数据库。在模拟仿真中，N 置为 80，L 置为 20，M 置为 30。这样的编码方法与二元神经网络是一致的(其神经元要么激活，要么不激活)，所有的激活信号均为二元表示。

在训练阶段，特征编码和类标签编码均出现在联想记忆中。该信息用来发掘处理元素的输入空间的潜在联系。在测试阶段，只有特征编码出现在输入层，而类标签编码被填充为未定义值。通过反馈机制，网络做出联想，并确定类标签的编码值。类标签编码值确定后，系统根据最小汉明距离为每个测试样本的类标签投票，以此编码所有可能类的值。分类精度是通过正确分类的样本数量与测试样本总数量的比率计算的。

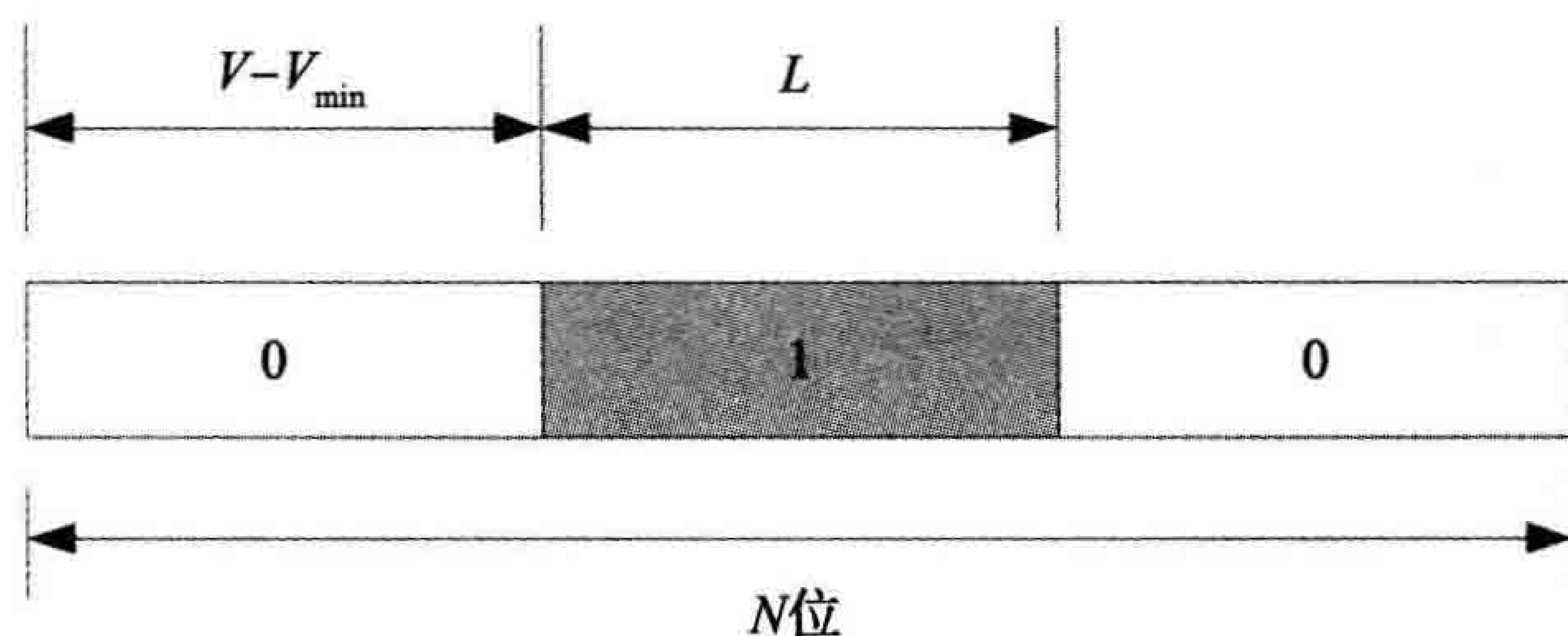


图 6-9 N 位输入的滑动条编码机制

由于在 Iris 数据库中只有 150 个实例，可用 K-fold 交叉检验方法来处理这些小样本数据集(Hong & Chen，2000；Dasarathy，1998；Quinlan，Compton，Horn & Lazarus，1987；Lee，Chen，Chen & Jou，2001；Chatterjee & Rakshit，2004)。在 K-fold 交叉检验方法中，所有样本被随机分成大小尽量相等的 K 个子

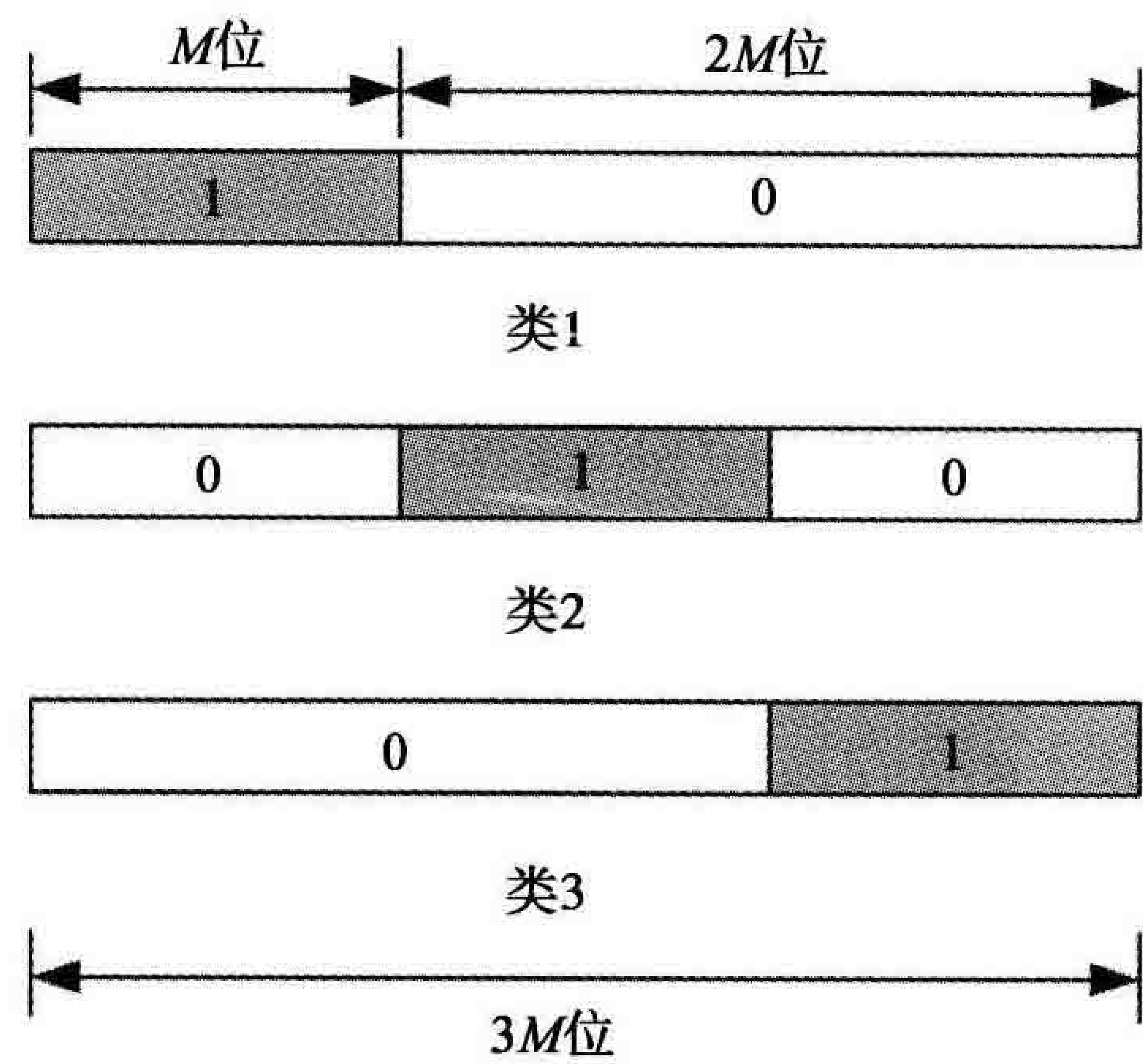


图 6-10 类标签的滑动条编码机制

集。在每次试验中，取 K 个子集中的一个作为测试集，其余 $K-1$ 个子集作为训练集。为了每个样本测试一次，K 次试验是必要的。最后的分类精度是计算所有 K 次试验的平均结果而来的。在模拟仿真中，K 置为 10(10-fold 交叉验证)。图 6-11a 说明了联想 PE 及其连接结构，图 6-11b 说明了联想 PE 对于部分网络的激活活动。其中，y 轴表示输入位，x 轴表示到输入的距离(联想深度)；圆圈表示联想 PE，图中也标记了反向传播路径。在输入层，实心圆表示正确识别的类标签编码位。可以注意到，在 Iris 数据库中学习联想，网络仅需要 6 层。

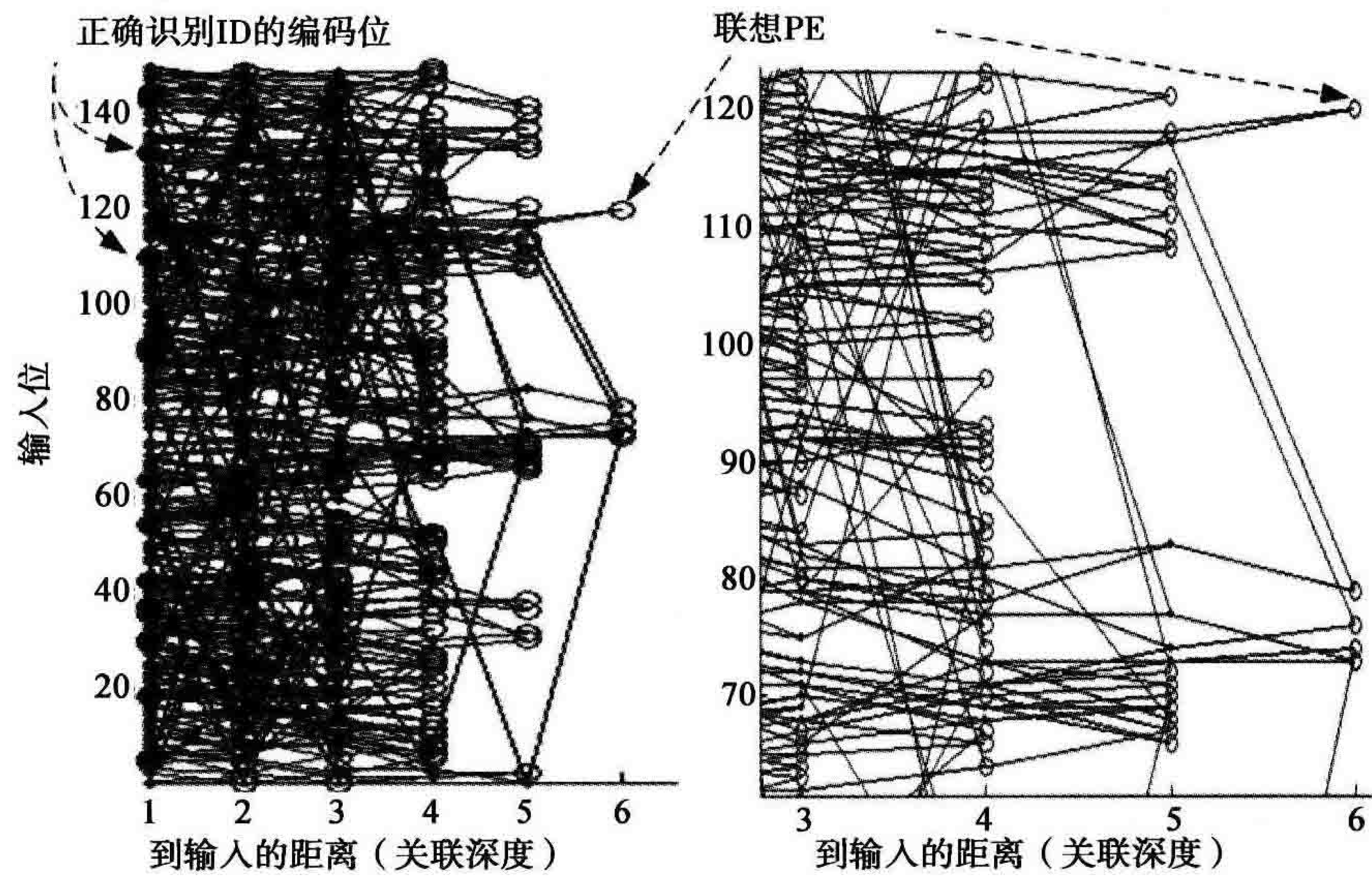


图 6-11 联想 PE 及其内部连接结构

表 6-3 给出了本章所描述模型的联想记忆分类性能与其他文献中所描述的分类

性能的比较(使用相同数据库)。这些结果表明，本章所提出的自组织联想记忆在分类问题方面具有令人满意的性能，这也就意味着通过联想可以进行自组织学习。在强化学习的价值体系中，如果想要关联特定的行为，通过联想进行学习是十分有用的。

表 6-3 Iris 数据库上的分类性能比较

方法	平均分类精度
合并成员权函数优先(Hong & Chen，2000)	97.33%
C4 方法(Hong & Chen，2000；Quinlan 等，1987)	93.87%
有影响力的规则搜索方案(Chatterjee &Rakshit，2004)	96.00%
Dasarathy 的模式识别方法(Hong & Chen，2000；Dasarathy，1980)	94.67%
基于模糊熵的模糊分类器(Lee 等，2001)	96.70%
自组织联想记忆(提出的方法)	96.00%

6.4.2 自联想应用

图像恢复问题可以用来测试所提出的记忆网络在自联想应用中的有效性。这对于某些应用是非常必要的：其中只有部分图像可用，而且没有具体说明其类别。利用无监督学习，所提出的记忆模型可以通过学习获得训练数据的特征，自主决定反馈深度，并进行联想恢复原始图像。

1. 熊猫图像恢复

本实例使用的是 Djuric、Huang 和 Ghirmai(2002)中的 64×64 二值熊猫图像。熊猫图像的第 i 行可以表示为向量 $p_i(=(x_1, x_2, \cdots, x_n))$，其中 $x_i=1$ 为黑色像素，$x_i=0$ 为白色像素。在测试中，在 p_i 中随机选取 $n\times r\%$($r=10$、20 或 30)个像素并将其设置为未定义值(0.5)，生成斑块图像。

原始熊猫图像和斑块图像分别如图 6-12a 和图 6-12b 所示。图 6-12c 给出了用所提出的联想记忆方法恢复的图像。正如在 Djuric 等(2002)中，图像的恢复性能通过计算错误恢复的像素数量(恢复后，错误恢复的像素和剩下的未定义像素)占总像素数量的比率来计算。从表 6-4 可以看出，相比 Djuric 等人提出的模型，自组织记忆模型具有较强的竞争力。

表 6-4 熊猫图像恢复错误信息

噪声水平	10%	20%	30%
参考(Djuric 等，2002)	2.95%	4.83%	6.57%
自组织联想记忆(提出的方法)	0.24%	0.39%	0.44%

图 6-12 说明了自组织联想记忆在随机斑块图像恢复中的应用，而图 6-13 展示

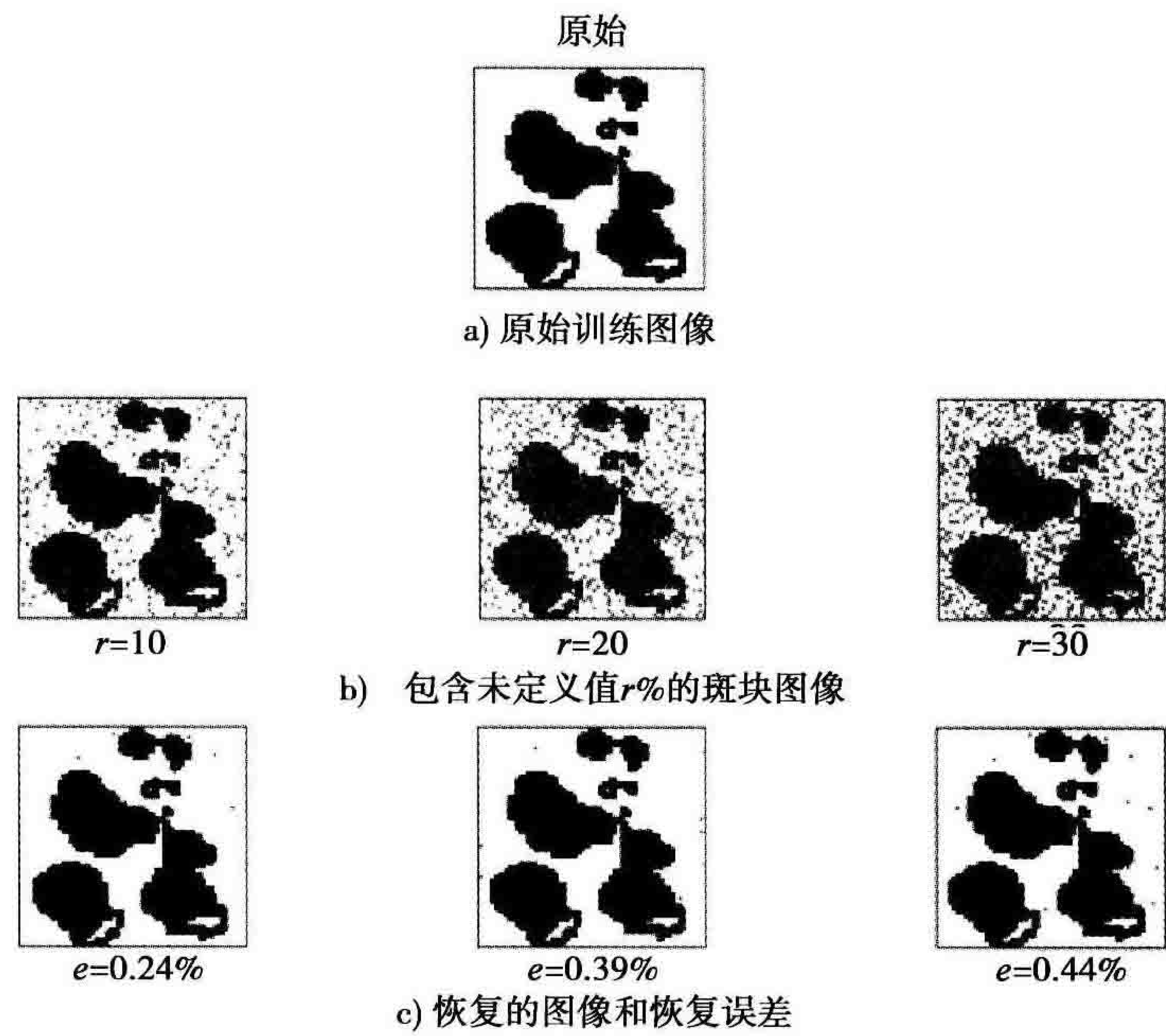


图 6-12 64×64 的二值熊猫图像

了熊猫图像在整个下半部分缺失的情况下的恢复性能。这种情况下，恢复错误比特占整幅图像的 2.42%。

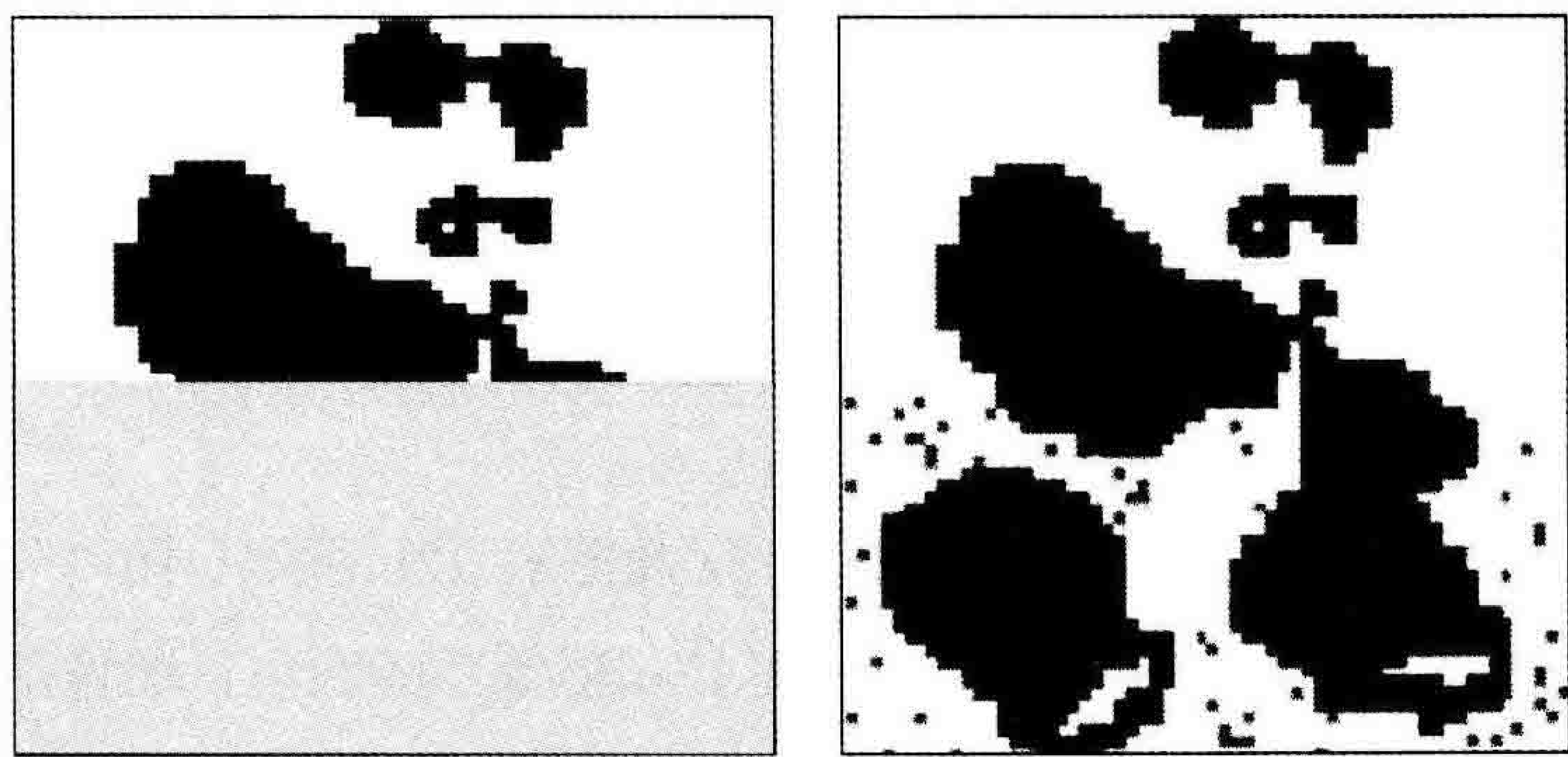


图 6-13 测试图像(半块)和恢复的图像

2. 汉字识别与恢复

根据 Wong 和 Chan(1998)所讨论的一些原因，汉字识别被认为是一个相当有挑战性的问题。首先，汉语的词汇量十分庞大；其次，许多汉字非常相似；最后，汉字本身很复杂。针对汉字识别，Wu 和 Batalama(2000，2001)提出了一种局部相同索引联想记忆算法。在 16 个汉字原型模式数据集上，上述方法的识别精度能达到 97.3%(Wu & Batalama，2000)和 100%(Wu & Batalama，2001)。Fu & Xu(1998)

针对多元语言手写汉字识别，提出了一种基于贝叶斯决策的神经网络模型(BDNN)。BDNN是一种自增长概率决策神经网络，采用层次网络结构，该网络结构具有非线性基函数和具有竞争性的信度分配方案。在3个不同汉字数据库上的仿真结果表明识别率达到86%～94%。现存的许多文献只处理汉字识别问题。由于所提出的自组织联想记忆模型具有分类和图像恢复的功能，本节将详细说明其在汉字识别与恢复方面的应用。

图6-14显示了5个黑白汉字(在每个汉字的顶部给出了相对应的英文)。正如我们看到的，这5个汉字非常相似。对每个汉字扫描后，可用向量的形式表示(20×20的图像)。与"熊猫图像恢复"小节一样，每个模式表示为向量 $p_i(=(x_1,x_2,\cdots,x_n)$，$n=400)$。其中 $x_i=1$ 为黑色像素，$x_i=0$ 为白色像素。在测试时，每个汉字被随机划分出50%，也就是说200个随机选择的像素将置为未定义值(0.5)。用提出的联想记忆模式进行识别和恢复，图6-15显示了输入的测试模式，相对应的训练模式如图6-14所示。



图6-14 训练模式：5个黑白汉字



图6-15 具有50%像素(200像素)的测试模式

下面用两种指标来评估联想记忆模型在这个实例中所表现出的性能。第一个是正确识别率，类似于6.4.1节中的定义。对于这些相似的汉字，10次仿真模拟的平均正确识别率为100%。第二个指标是观察所恢复的汉字像什么，错误比特和丢失比特的比率是多少，类似于"熊猫图像恢复"小节。

图6-16显示了与图6-15相对应的恢复模式，表6-5给出了每个模式的错误比特和丢失比特信息。图6-16的灰色部分显示了恢复后仍丢失的像素，从图6-16可以看出，即使只有一半的像素，联想记忆模型也可以正确地恢复原始图像。表6-5

给出了平均错误比特和平均丢失比特的比率分别为6.2%和2.4%。

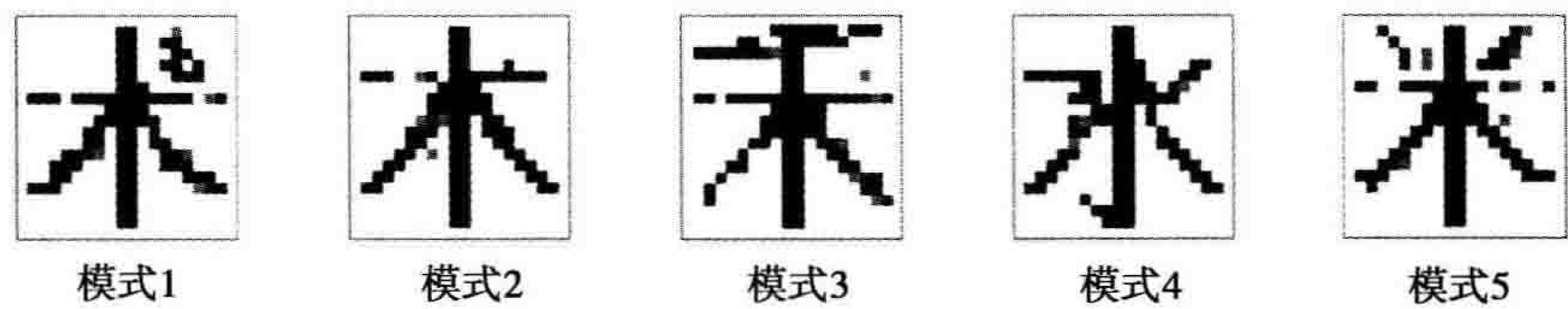


图6-16 测试模式恢复的汉字

表6-5 测试中错误比特和丢失比特信息

测试模式	模式1	模式2	模式3	模式4	模式5	均值
错误比特	12	9	14	13	14	12.4
错误比特率	6%	4.5%	7%	6.5%	7%	6.2%
丢失比特	6	5	7	2	4	4.8
丢失比特率	3%	2.5%	3.5%	1%	2%	2.4%

3. 在线增量学习的联想记忆

正如本书第2章的讨论，增量学习对于开发自适应智能系统相当重要。所提出的联想记忆也具有增量学习的能力。事实上，为了更清楚地描述，上述例子中的训练阶段与测试阶段的差异是人为设定的。在记忆框架中的所有PE均可以不断学习，积累知识，并用流数据进行预测。图6-17说明了这种记忆从26个字符(A到Z)不断学习的应用。所有这些模式连续地出现在具有不同噪声水平的记忆模型中，记忆模型能够随着时间推移不断学习，同时根据所积累的知识进行预测。从图6-17可以看出，随着数据流的连续输入，记忆模型可以有效观测到原始数据中的信息，更新其内部的知识表达，并在PE的分布式网络中建立联想，以支持决策和预测过程。

总之，本章所呈现的各种应用问题的仿真结果表明，所提出的自组织联想记忆模型，无论是异联想还是自联想都具有优越的性能。我们不能预料所提出的模型在同类方法中是最好的，但对于特定问题却是最优的(例如，固定类数的分类问题)。因其不改变网络结构就能解决各种问题，所以所提出的记忆模型具有较强的鲁棒性。为此，我们认为一个灵活的结构比专门的和固定的结构更重要，这种灵活的结构能够积累知识，利用所积累的知识达到更高水平的自组织并解决问题。本节所提出的联想记忆模型的目标是寻找一种网络结构，它可以扩展到建立真正的智能机器，也可能扩展到类似生物学研究所关注的“巨柱”和“微柱”结构(Mountcastle，1997；Jones，2000)。

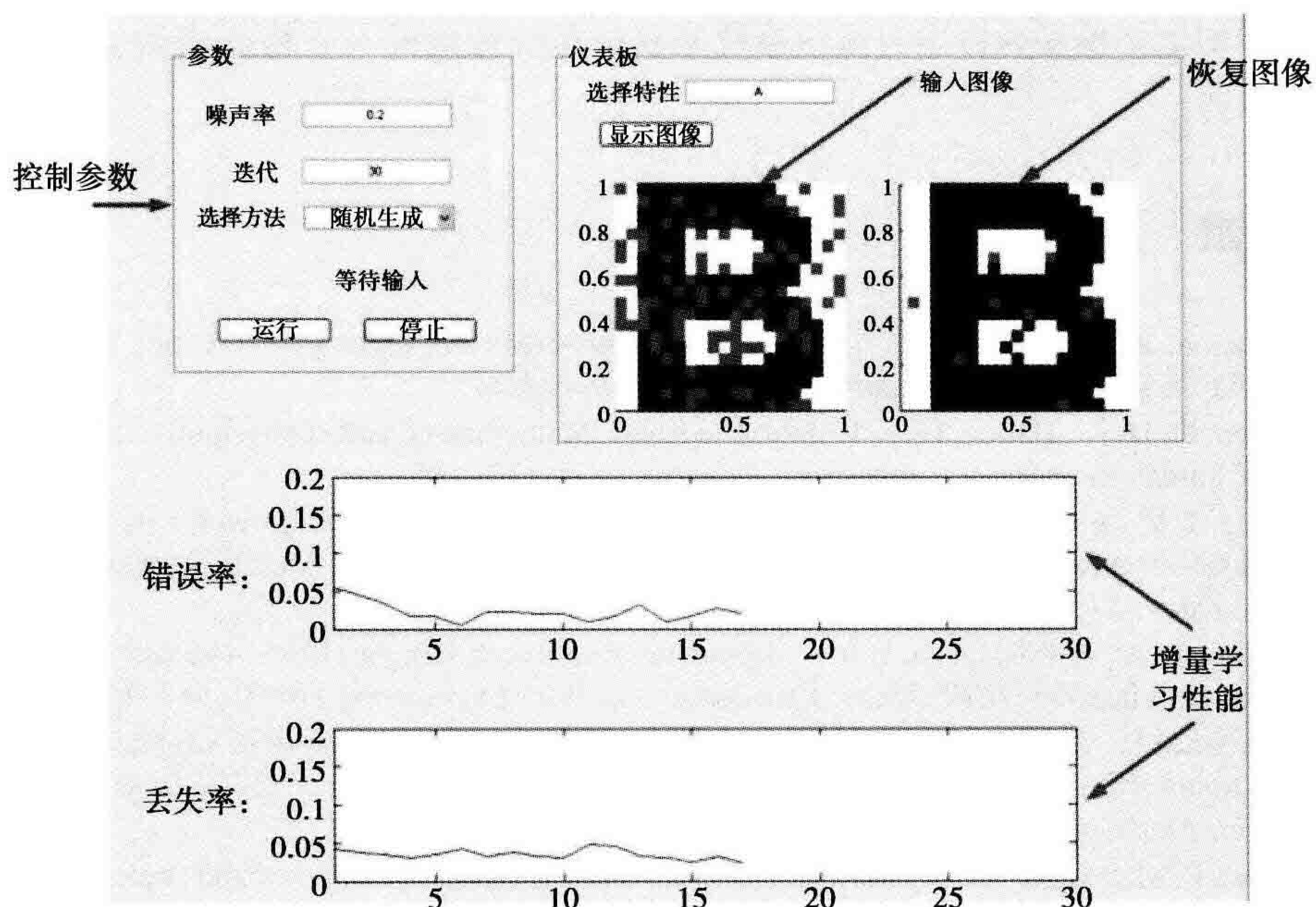


图 6-17 联想记忆的增量学习能力示例

6.5 总结

本章介绍了机器智能中的自组织联想记忆，并研究了异联想和自联想在不同领域中的应用。本章要点包括：

- 自组织和联想是学习框架所期望的特性，并在类脑智能发展中发挥着重要作用。自组织结构不需要明确的监督，并选择性地影响学习网络的不同区域，它依赖于如何将新数据与存储在网络中的信息进行关联以及新兴网络行为如何对机器学习产生影响。人类记忆的自组织在不同感官与外部世界的内在表示之间建立自然的联想。
- 正如生物智能系统，在所提出的自组织记忆网络中不存在预先设定的能表示一个特定的概念或建立一种特定联想的 PE。这些都是从传感器、PE 和外部世界之间的相互作用中自发产生的。特别是，在这种网络中，信号的传播深度是不可预知的，它是根据来自外部环境的传感器输入自主确定的。可以观察到，在更复杂的情况下，联想的深度越深，决策过程涉及的 PE 越多。
- 人类大脑能够进行异联想和自联想。所提出的记忆模型通过使用有效的概率推断算法以及前馈机制、反馈机制，也能够进行异联想和自联想。因此，这

样的记忆模型为开发自适应系统来获取类脑智能的本质特征提供了一个重要手段。

参考文献

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository [Online], Available: http://www.ics.uci.edu/mlearn/MLRepository.html*.

Brown, G. D. A., Dalloz, P., & Hulme, C. (1995). Mathematical and connectionist models of human memory: a comparison. *Memory*, *3*(2), 113–145.

Chang, J. Y., & Cho, C. W. (2003). Second-order asymmetric bam design with a maximal basin of attraction. *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, *33*, 421–428.

Chatterjee, A., & Rakshit, A. (2004). Influential rule search scheme (IRSS)—a new fuzzy pattern classifier. *IEEE Trans. Knowledge and Data Engineering*, *16*(8), 881–893.

Dasarathy, B. V. (1980). Noising around the neighborhood: a new system structure and classification rule for recognition in partially exposed environments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *2*(1), 67–71.

Djuric, P. M., Huang, Y., & Ghirmai, E. (2002). Perfect sampling: a review and application to signal processing. *IEEE Trans. Signal Processing*, *50*(2), 345–356.

Fu, H. C., & Xu, Y. Y. (1998). Multilinguistic handwritten character recognition by bayesian decision-based neural networks. *IEEE Trans. Signal Processing*, *46*(10), 2781–2789.

Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: Times Books.

Hawkins, J., & Blakeslee, S. (2007). Why can't a computer be more like a brain? *IEEE Spectrum*, *44*(4), 20–26.

Hong, T. P., & Chen, J. B. (2000). Processing individual fuzzy attributes for fuzzy rule organizing learning arrays. *Fuzzy Sets and Systems*, *112*, 127–140.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Sciences of the USA*, *79*, 2554–2558.

Jones, E. G. (2000). Multicolumns in the cerebral cortex. *Proc. National Academy of Sciences of the USA*, pp. 5019–5021.

Lee, H. M., Chen, C. M., Chen, J. M., & Jou, Y. L. (2001). An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics*, *31*(3), 426–432.

Malsburg, C. V. (2003). *Handbook of brain theory and neural networks*. Cambridge, MA: MIT Press.

Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, *120*, 701–722.

Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (todam2). *Psychological Review*, *104*, 839–862.

Quinlan, J. R., Compton, P. J., Horn, K. A., & Lazarus, L. (1987). Inductive knowledge acquisition: A case study. *Proc. Australian Conf. Applications of Expert Systems*, pp. 137–156.

Rizzuto, D. S., & Kahana, M. J. (2001). An autoassociative neural network model of paired-associative learning. *Neural Computation*, *13*, 2075–2092.

Salih, I., Smith, S. H., & Liu, D. (2000). Synthesis approach for bidirectional associative

memories based on the preceptron training algorithm. *Neuralcomputing*, *35*, 137–148.

Starzyk, J. A., He, H., & Li, Y. (2007). A hierarchical self-organizing associative memory for machine learning. *Lecture Notes in Computer Science*, *4491*, 413–423.

Starzyk, J. A., & Wang, F. (2004). Dynamic probability estimator for machine learning. *IEEE Trans. Neural Networks*, *15*, 298–308.

Starzyk, J. A., Zhu, Z., & Li, Y. (2006). Associative learning in hierarchical self organizing learning arrays. *IEEE Trans. Neural Networks*, *17*(6), 1460–1470.

Triesch, J. (2004). Synergies between intrinsic and synaptic plasticity in individual model neurons. *Advances in Neural Information Processing Systems*, *17*.

Vogel, D., & Boos, W. (1997). Sparsely connected, hebbian networks with strikingly large storage capacities. *Neural Network*, *4*(10), 671–682.

Wang, L. (1999). Multi-associative neural networks and their applications to learning and retrieving complex spatio-temporal sequences. *IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics*, *29*, 73–82.

Wang, M., & Chen, S. (2005). Enhanced emam based on empirical kernal map. *IEEE Trans. Neural Networks*, *16*, 557–563.

Wong, P. K., & Chan, C. (1998). Off-line handwritten chinese character recognition as a compound bayes decision problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *20*(9), 1016–1023.

Wu, Y., & Batalama, S. N. (2000). An efficient learning algorithm for associative memories. *IEEE Trans. Neural Networks*, *11*(3), 1058–1066.

Wu, Y., & Batalama, S. N. (2001). Improved one-shot learning for feedforward associative memories with application to composite pattern association. *IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics*, *31*, 119–125.

第7章

序列学习

7.1 引言

序列学习也许是人类智能中最重要的组成部分之一，因为人类的大多数行为是序列形式的，包括但不限于自然语言处理、语音识别、推理和计划等(Sun & Giles，2001；Anderson，1995；Schneider & Logan，2006)。在过去的几十年中，序列学习的模型和机制已经引起了极大的关注。例如，McClelland 等人开创性地研究了序列学习的并行分布式处理(PDP)模型及其在语音与语言处理方面的应用(McClelland & Rumelhart，1981；Seidenberg & McClelland，1989；Plaut，McClelland，Seidenberg & Patterson，1996；McClelland & McClelland & Elman，1986a，1986b)，如交互激活模型(McClelland & Rumelhart，1981)、TRACE 模型(McClelland & Elman，1986a，1986b)等。最近，Sun 和 Giles(2001) 对序列学习的特性、存在的问题和挑战给出了重要综述，范围涵盖了从序列识别、预测到序贯决策。本章针对复杂序列的学习、存储和检索(Starzyk & He，2007)提出了一种基于预测的分层神经网络结构。

7.2 序列学习的基础知识

首先回顾序列学习的相关基础知识，特别是以下重要特性：自组织、分层结构、时间表征、预期和联想能力。

自组织对序列学习模型中分布式内部信息表示的设计及其与非结构化、不确定环境主动交互的关联至关重要。自组织模型已广泛应用于序列学习的研究中。例如用于双语信息处理的自组织连接模型(Li & Farkas，2002a，2002b，2002c；Farkas & Li，2007)。Li & Farkas(2002a)提出的 SOMBIP 模型用两个内部相连的具有回归连接的自组织神经网络计算词汇同现的约束条件。DevLex(DevLex-Ⅱ)模型是另

一个基于自组织映射的神经网络模型，用于语言采集(Farkas & Li，2002；Li & Farkas，2002b，2002c；Zhao & Li，2007)。DevLex模型的核心思想是用增长语义映射(GSM)和音韵映射(PMAP)通过相互连通的联想路径设计字典。进化算法也可以被集成到自组织神经网络，以学习序列决策任务和语言处理(Miikkulainen，1990，1992，1993；Moriarty & Miikkulainen，1999；James & Miikkulainen，1995)。例如，一个名为DISCERN的自然语言处理系统在亚符号级上发展起来了(Miikkulainen，1990，1993)。DISCERN系统的关键特性之一是使用层次化组织的特征映射和模块进行信息处理。Moriarty和Miikkulainen(1999)提出的SANE方法是一种基于遗传算法和神经网络的共生、自适应神经元进化模型。SANE的核心思想是通过遗传算法使神经网络进化，从而在更宽泛的领域内用最小的强化来学习。最近，心理学研究也对序列学习的事件感知和时间处理提出了重要建议(Lewkowicz，2004，2006；Lewkowicz & Marcovitch，2006；Lewkowicz & Ghazanfar，2006)。例如，用不同的实验研究婴儿对序列命令的感知、学习和辨别能力(Lewkowicz，2004)。Lewkowicz和Marcovitch(2006)研究了不同年龄段的婴儿对复杂视听节奏模式的感知能力，实验表明，4～10个月的婴儿不仅可以感知和辨别复杂的视听时间模式，还能够学习这些模式的不变性质。对多个连续阵列的视觉短时记忆(VSTM)现也有不同的研究(Kumar & Jiang，2005；Liu & Jiang，2005；Chun & Jiang)。例如，Jiang等(2000)的研究表明，VSTM的组织结构包括与全局空间结构相关的单个视觉项的信息，这为发展有效的时空记忆模型提供了重要的思想。

层次结构被认为是高效序列学习、存储和检索的一个重要特性(Starzyk & He，2007；Schneider & Logan，2006；Wang & Arbib，1993；Manning & Witten，1998；Hawkins & Blakeslee，2004，2007；Hawkins & George，2006；Wang & Yuwono，1996)。例如，Schneider和Logan(2006)设计了4个实验，通过研究认知过程的分层处理来检测连续行为中的序列与任务层处理之间的关系。Wang等人的一系列工作表明序列可以用分层组织有效地学习、预测和检索(Wang & Arbib，1990，1993；Wang & Yuwono，1995，1996；Wang & Arbib，1990)。例如，Wang和Arbib(1990)提出了两种特殊类型的神经元：第一种是被用来存储短时信号的双神经元，它输出分级信号。这与许多神经网络模型中传统的二元信号不同。第二种是序列检测神经元。经过学习，这种序列检测神经元能被以前的模式序列激活，而不仅仅是以前的模式，从而克服了网络不能可靠地回忆起有相同模式的序列的限

制。基于这项工作，Wang 和 Arbib 提出了一种学习、识别和复制复杂时间序列的框架。在这种模型中，序列是通过注意力学习规则获得的，该规则结合了 Hebbian 学习规则和顺序系统激活的归一化规则。序列分量之间的时间间隔不影响识别。全局抑制的提出能够使模型通过学习所需的上下文长度来消除复杂序列复制中的歧义。为了克服短时记忆(STM)的容量限制，Wang 和 Arbib 讨论了一种基于分块机制的分层序列识别模型。例如，在一个字母-单词-句子的分层结构中，一个给定单词的单元是在该单词表述结束时被激活的，然后模型根据当时被激活的字母单元来学习该单词中字母的序列。一旦学会了单词的结构，就可以用相同机制训练更高层次的单词序列。Wang 和 Arbib 解决的另一个问题是间隔保持，它是通过对检测层到输入层的间隔连接权值进行编码实现的。此外，Wang & Yuwono(1995)提出了一种能够通过自组织学习并产生复杂时间模式的神经网络模型，这种模型能够积极地再生成序列的下一个组件，并将预期的组件与下一个输入进行比较，通过模型预测与实际输入之间的误匹配触发单次学习。Wang 和 Yuwono(1996)讨论了复杂时域模式和灾难性干扰的增量学习问题。复杂序列增量学习中的灾难性干扰主要是由于改变存储模式的权值所导致的。人类记忆中虽然可能会出现一些干扰(Bower，Thompson-Schill & Tulving，1994)，但不存在灾难性的干扰。显然，Wang 和 Yuwono(1995)提出的预测模型可能对增量学习有逆向干扰，但没有灾难性干扰。此外，此模型还提出了用来检测序列之间和序列内重复子序列的分块机制，以充分减少保留在连续训练中的序列数量。Manning 和 Witten(1998) 提出了一种名为 SEQUITUR 的线性时间算法，用来识别序列的分层结构。这个算法的主要思想是：不止一次出现的短语会由产生该短语的语法规则代替，并且这个过程会连续递归，最终产生原始序列的一个分层表示。Geroge 和 Hawkins(2005)讨论了不变模式识别的时间序列学习的分层结构问题。结果表明，新大脑皮质解决了分层结构中的不变性问题：层次化结构中每个区域学习并召回输入序列，每一层的时间序列成为下一个更高层区域的空间输入。最近，Hawkins 等给出了分层结构是机器智能研究的关键的主要原因(Hawkins & Blakeslee，2004，2007；Hawkins & George，2006)：泛化能力和存储效率(共享表示)，与现实世界中空间、时间层次的一致性，快速响应及内隐性注意。

序列学习的另一重要问题是时间表示，其难点在于如何用一种自然的并且在生物学上可行的方式来表示时间(Elman，1990)。因此，递归神经网络成为研究序列学习问题的有力工具(Elman，1990；Jordan，1986；Pollack，1991；Jacobsson，

2005)，且已有许多成功应用，包括手臂机器人的连续行为学习和机器人导航系统(Tani & Nolfi，1999；Tani，2003)。但是，使用随时间反向传播(BPTT)算法的传统递归神经网络无法在长时间延迟后学习序列(Hochreiter & Schmidhuber，1997)。这是因为反向传播错误信号会在BPTT机制中消失。为了克服这个限制，Hochreiter和Schmidhuber(1977)提出了结合基于梯度的学习算法的递归网络体系结构，即长短时记忆网络(LSTM)。其主要思想是在递归神经网络中引入一个存储单元(常在误差传输)，因此，误差信号不会消失，系统能够长时间、连续动态地学习。

最后，预测和联想能力对于序列学习、存储和检索也很重要(Starzyk & He，2007；Wang & Arbib，1990，1993；Wang & Yuwono，1995，1996)。通过正确预测，智能系统表明它已经知道这些知识，因此无需额外学习。对于这种方式，仅仅在预测错误时才需要学习。例如，Wang和Yuwono(1995)提出了一种预测机制，通过匹配当前的输入与预测的信息来实现单次学习。Ara'ujo和Barreto(2002)对以预估方式进行序列学习的神经网络模型进行了研究，并成功将其应用于机器人路径规划中。联想学习机制也能够用于复杂序列的学习和预测中。例如，Wang(1998，1999)研究了具有延迟反馈连接的联想神经网络在序列学习和检索中的应用。特别地，Wang(1998)针对时空序列学习提出了一种联想记忆模型。该模型具有3个主要组成部分：投票网络、异联想神经网络(HANN)的并行阵列，以及从系统输出到联想神经网络层的延迟反馈线路。异联想的延迟序列在每个时间步长下对下一个输出进行“投票”。经过学习之后，系统可以从一个小的提示序列检索整个序列。由于Wang(1998)中的模型假设每个异联想神经网络只学习单一空间模式间的异联想，而不学习群组内多个模式间的联想，所以该模型被进一步扩展到包括一个模式和多个模式之间的联想。Wang(1999)提出的模型具有学习时间短、检索精度高的优点，并且能够存储大量的复杂序列。对连续记忆使用联想记忆方法的最新发展包括动态异联想记忆(Chartier & Boukadoum，2006)、基于上下文层和移位寄存器模型的联合模型(Bose，Furber & Shapiro，2005)、自组织联想记忆(Starzyk & He，2007)等。

7.3 分层神经结构的序列学习

本章重点讨论智能系统中多个复杂序列的学习、存储和检索。在这里，采用Wang和Arbib(1990，1993)中关于序列定义的术语：

$$S:U_1-U_2-\cdots-U_k \tag{7-1}$$

其中，$U_i,i=1,\cdots,k$ 为序列 U_i 的一个分量或符号(例如一个字符、一个数字或其他)，序列 S 的长度为 k。一般来说，一个序列可能包含同一子序列在不同上下文时的多次重复，例如 $S1$：$A-S-C-D-B-C-E$。这样，需要复制 S 中当前符号 U_i 的上下文被定义为 U_i 前的子序列，其长度定义为 U_i 的度。例如，在序列 $S1$ 中，符号 D 的度是 3。用这种方式，我们可以把序列的度定义为所有分量的度的最大值。这样，度为 1 的序列称为简单序列，其余称为复杂序列(Wang & Arbib，1990，1993)。图 7-1 为一个具有 3 级序列表示的分层组织：字母-单词-句子。

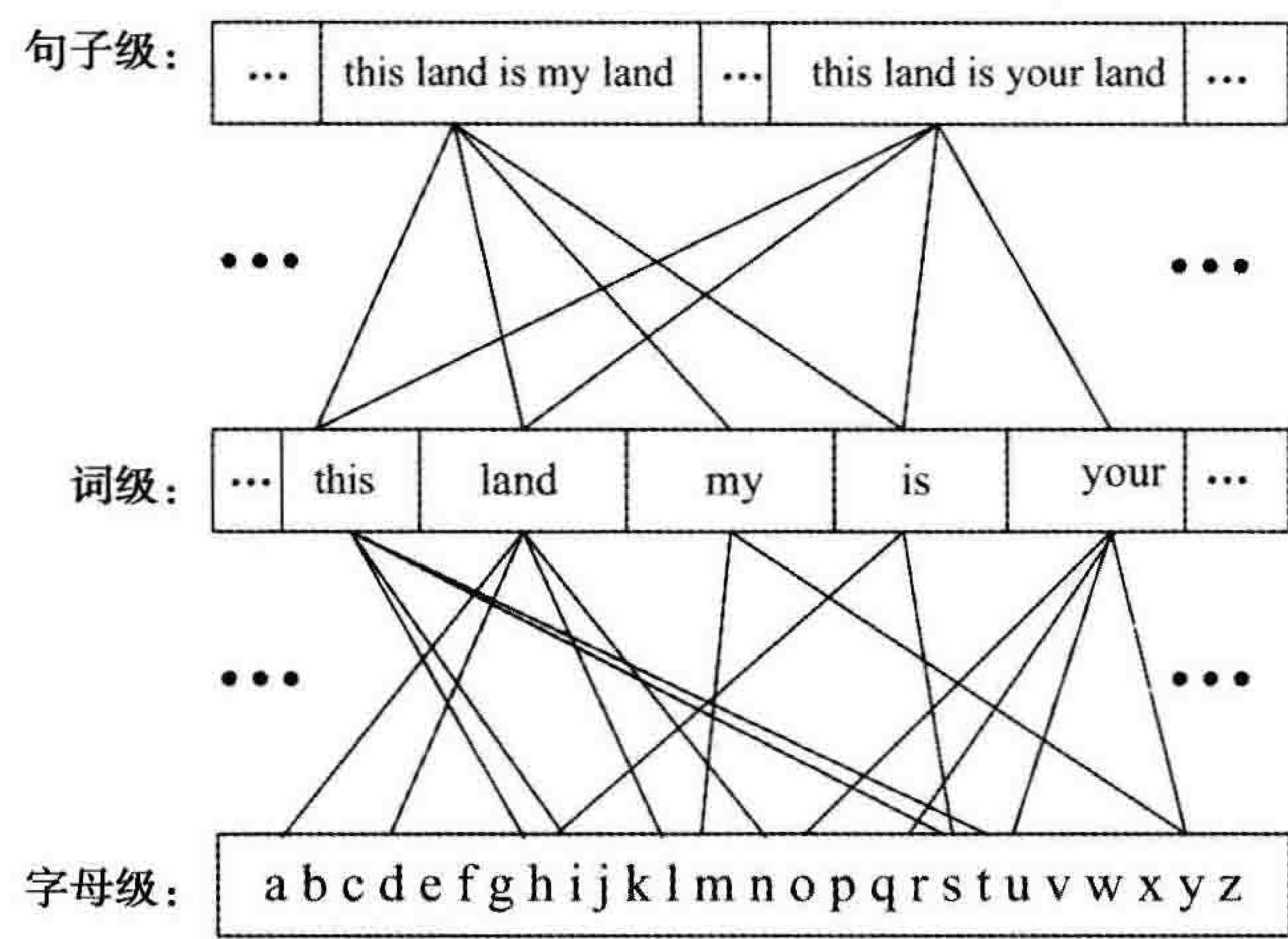


图 7-1　序列的分层组织

图 7-2 举例说明了所提出模型的分层组织的整体系统架构(Starzyk & He，2007)。在这个模型中，一层的输出是下一层的输入。每一层用胜者为王机制来选择活跃的神经元。第 1 层(0 层)用一种改进的 Hebbian 学习机制进行模式识别。对于序列学习，第 1 层到第 N 层的结构完全相同。每层的关键部分为输入寄存器(IR)神经元、时控多路复用器(MUX)、预测神经元(PN)、预测检查神经元(PCN)、预测匹配神经元(PMN)、学习标记神经元(LFN)、多个获胜者检测神经元(MWDN)和学习神经元(LN)。IR 神经元对下一较低层的一系列输出序列进行空间编码。这个序列可能是新加入的或者是从顺序记忆中召回的。MUX 顺序地调用 IR 的内容并与下一较低层输出的序列进行比较。这种组织结构对复杂序列的学习、存储和检索非常有效。

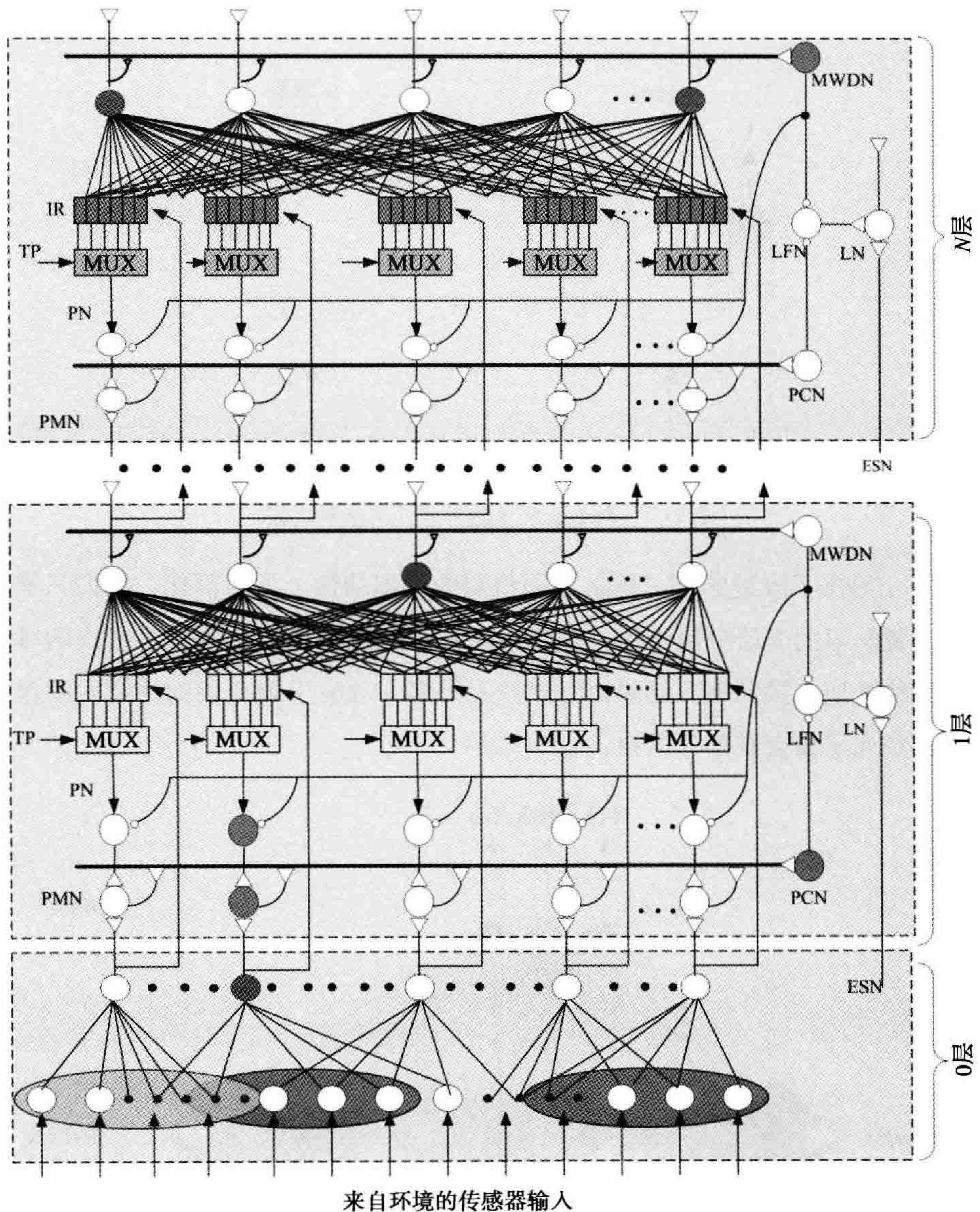


图 7-2 基于预测的分层序列学习模型的系统架构

7.4 0层：改进的 Hebbian 学习架构

在本节所描述的模型中，第1层使用改进的 Hebbian 学习机制(如图 7-2 中的0层)。考虑到生物神经元要么是激活的，要么未被激活，因此可假设来自环境的每个传感器输入为0或者1。图 7-3 举例说明了传感器信息的0-活跃和1-活跃表示的意义。当传感器输入为1时，左边的神经元被激活；当传感器输入为0时，输入值经过“逆变器”使右边神经元激活。通过这种方式，不同的神经元激活代表了不同传感

器输入的二元编码。

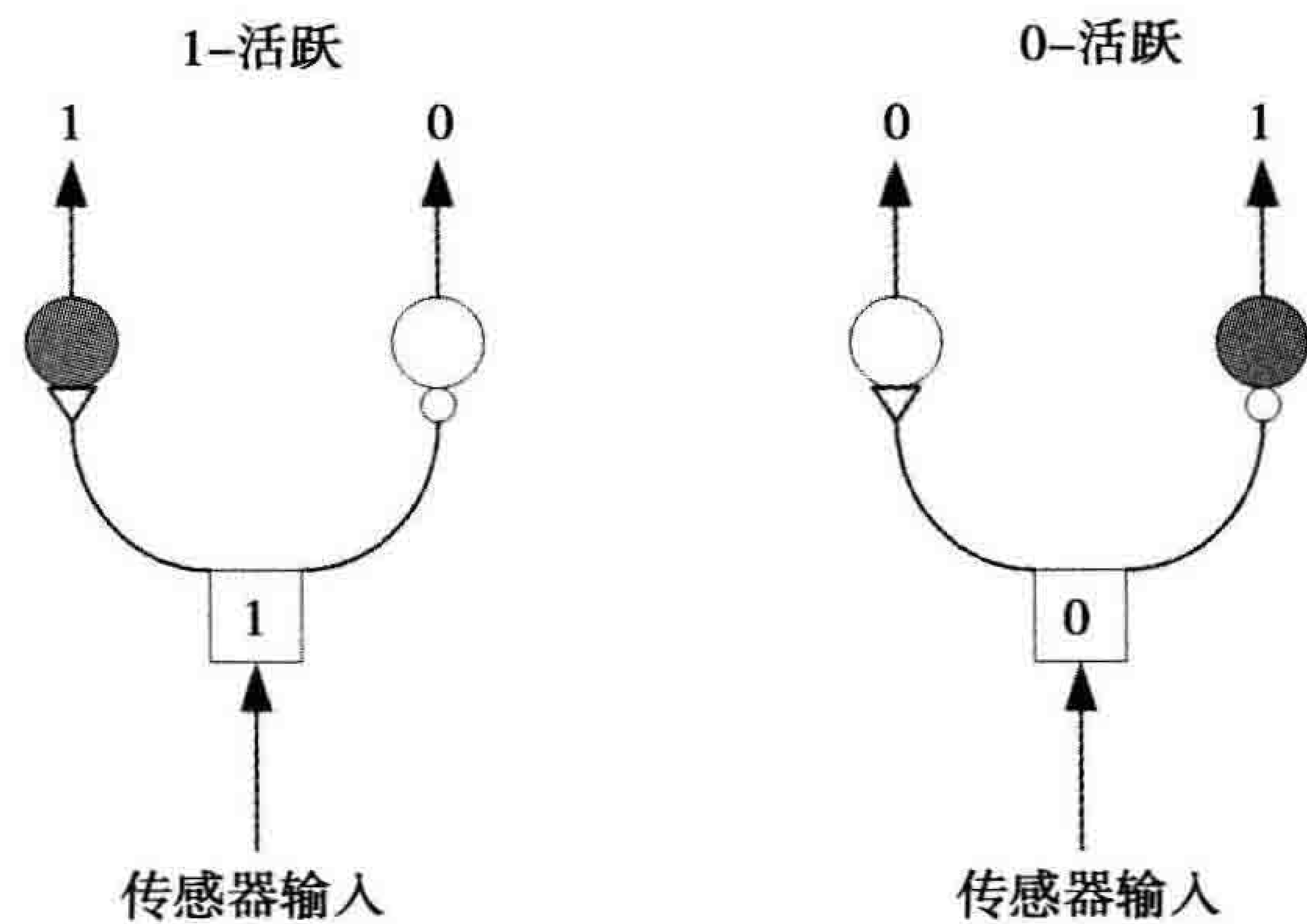


图 7-3 两个活跃区域神经元的激活机制

图 7-4 给出了改进的 Hebbian 学习机制的详细结构。为了简化，我们只展示了一个无监督学习的 3 层分层结构。第 2 层的神经元被分成几个组，每组内的每个神经元随机稀疏地连接到第 1 层的相同神经元子集。对应于第 2 层的神经元组投影的第 1 层神经元子集会有部分重叠。

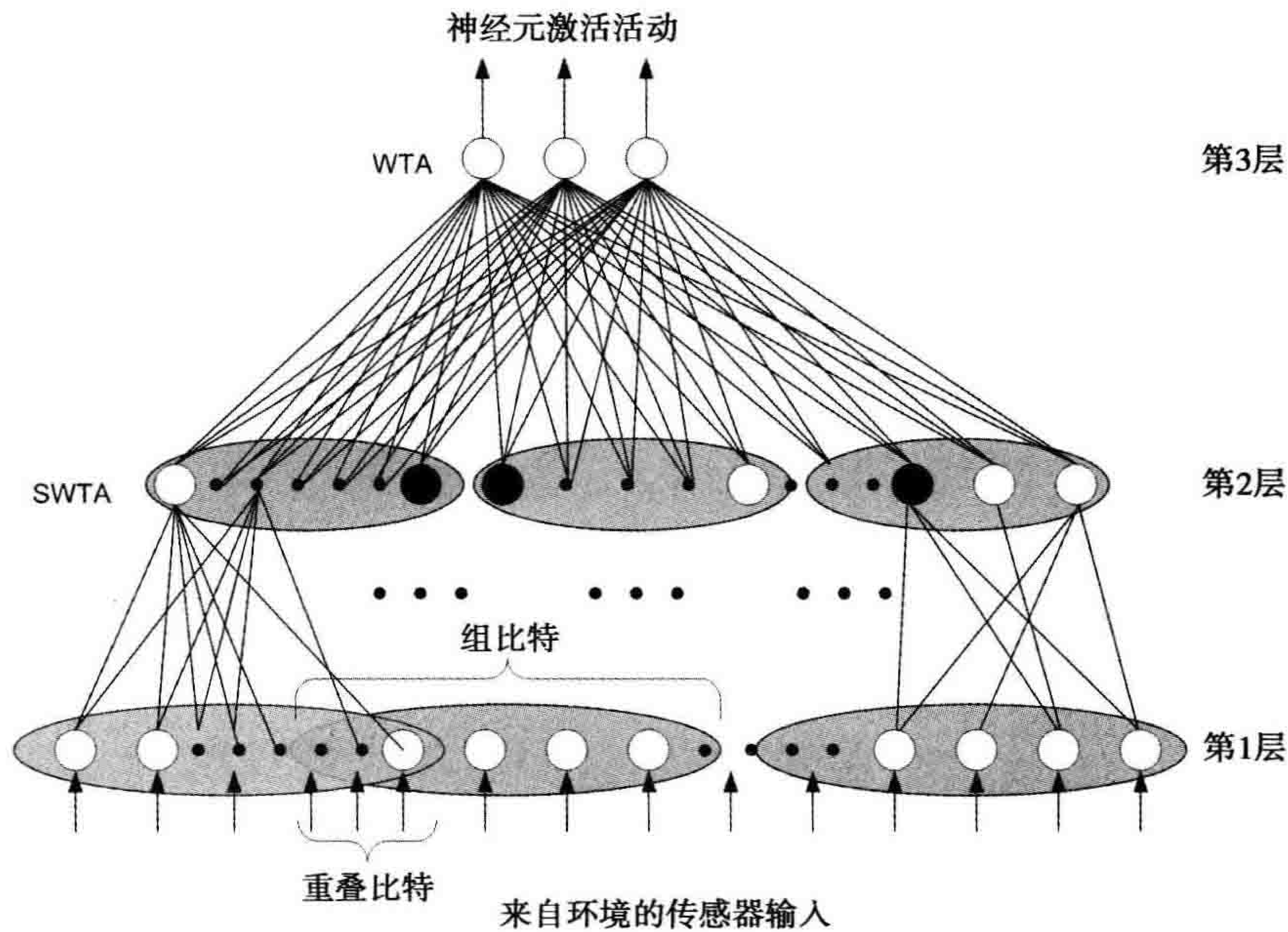


图 7-4 改进的 Hebbian 学习模型的分层结构

这个模型使用两种相似的 WTA 机制来提高模型的学习效率，降低学习的复杂度。第 1 种是刚性 WTA（SWTA），可以用一个简单的计数器实现，SWTA 用于第

2层。因为来自环境的传感器输入为0或1，SWTA只统计每个神经元接收到1的数量，并从每个分组中选择接收到1最多的神经元为获胜者。特别地，SWTA不调节权值。

第2种WTA机制用于输出层(见图7-4中的第3层)。首先，所有输出层神经元的权值 w_i 按照下列条件随机设置：

$$w_{ni} = \pm 1 \text{ 且} \sum_i w_{ni} = 0 \tag{7-2}$$

其中，n 表示神经元，$i=1,2\cdots,k$ 表示所有先前层的神经元到神经元 n 的连接。获胜者为：

$$\text{获胜者}(w) = \max_n(\sum_i w_{ni} I_{ni}) \tag{7-3}$$

其中，I_{ni} 表示 w_{ni} 出现时神经元的活跃度(0或1)。每一时间步长下，选定获胜者后，获胜者的权值 w_{wi} 下标 w 表示获胜者)调整如下：

接收输入 $I_{li}=1$ 时，连接为

$$w_{wi}(t+1) = w_{wi}(t) + \eta \times M(0) \tag{7-4}$$

接收输入 $I_{li}=0$ 时，连接为

$$w_{wi}(t+1) = w_{wi}(t) - \eta \times M(1) \tag{7-5}$$

其中，η 是一个小调整(学习率)，$M(1)$ 和 $M(0)$ 分别是神经元接收到1和0的数量。这种调整保证了获胜者在调整后的权值总和仍然为0。所有获胜者的权值调整后，将会线性收敛到区间[−1，1]内。无监督学习应用于这个模型，意味着无论何时出现新的训练样本，每个输出神经元都会更新它们的活跃度。一般地，第1层(0层)对序列 S 中的分量 $U_i, i=1,2,\cdots,k$ 进行识别。应该指出的是，也可以用其他机制进行0层的设计，为复杂序列学习中的较高层次提供输入表示。

7.5 1～N层：序列存储、预测和检索

7.5.1 序列存储

从现在开始，我们将重点放在如图7-2所示的1～N层，以说明该模型是如何对复杂序列进行学习、存储和预测的。在这个分层结构中，某一层的输出被存储在下一个更高层的IR中。IR中的神经元通过可训练的连接投射到同层的输出神经元上。

首先，所有的输出神经元都能通过与未经训练的电突触(电阻式)连接，从而完全连接到所有的IR。在现有模型中，所有输出神经元的初始化权值都被设定为小正

数 $0.001<w_i<0.01$。

一旦输入序列被存储在 IR，所有输出神经元进行竞争并且获胜神经元的权值会被调整。所有活跃神经元映射到 IR 中获胜神经元的权值被设定为 1(兴奋)，而所有其他映射到获胜神经元的权值被设定为 −100(抑制)。采用这种强抑制作用来保证：一旦一个输出神经元被用来存储一个序列，那么它将不再进行进一步的学习。例如，考虑如图 7-5 所示的 IR 状态，用三角形的链接表示兴奋映射，用圆形的链接表示抑制映射。各个不同 IR 的位置确定了这个神经元存储的序列“miss” 的字母顺序(即，在字母“m”上面有一个兴奋链接，位于 IR 神经元片段的第一时间步长上)。一旦一个输出神经元的链接被训练后，那么这个神经元仅对特定的输入序列产生响应。

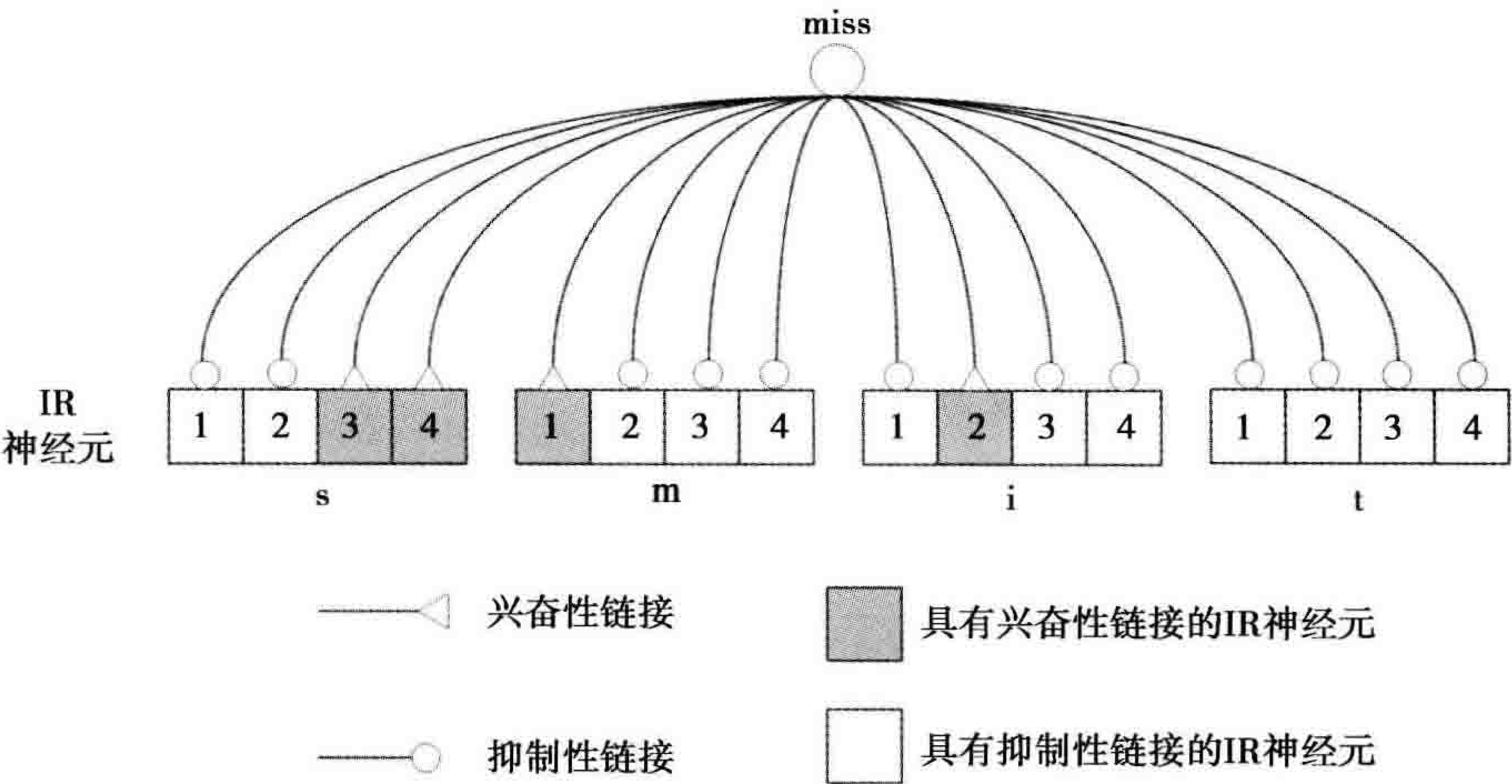


图 7-5 序列在 IR 神经元中的存储

IR 的结构如图 7-6 所示，其中包含指针神经元和具有抑制与兴奋链接的 IR 神经元。神经元间链接激活的单位延迟 Δ 在此研究中也有所考虑。

图 7-7 在 IR(见图 7-6)的基础上，举例说明了通过神经元活动时控的输入数据的存储情况(Starzyk & He，2007)。假设输入的序列数据是“AB”。在一个新序列的开始会出现“Start”启动信号(启动信号是新序列开始的脉冲信号)。“Start”信号通过抑制链接清除 IR 中的所有信息。同时，PT0 激活。当第一个数据“A”被接收，会产生“Next”信号。当“Next”信号变低时，移除上一个指针神经元的抑制。因此，延迟 Δ 时间后，PT1 会被激活。

再经过一次时间延迟，下一个指针神经元 PT2 从 PT1 中获得兴奋链接而被激活。同时，PT1 对 PT3 起抑制作用。如图 7-6 所示，由于“Start”信号变低且 PT1 和输入数据神经元是活跃的，所以 IR 神经元 IR1 被激活，并将第一个数据存储到

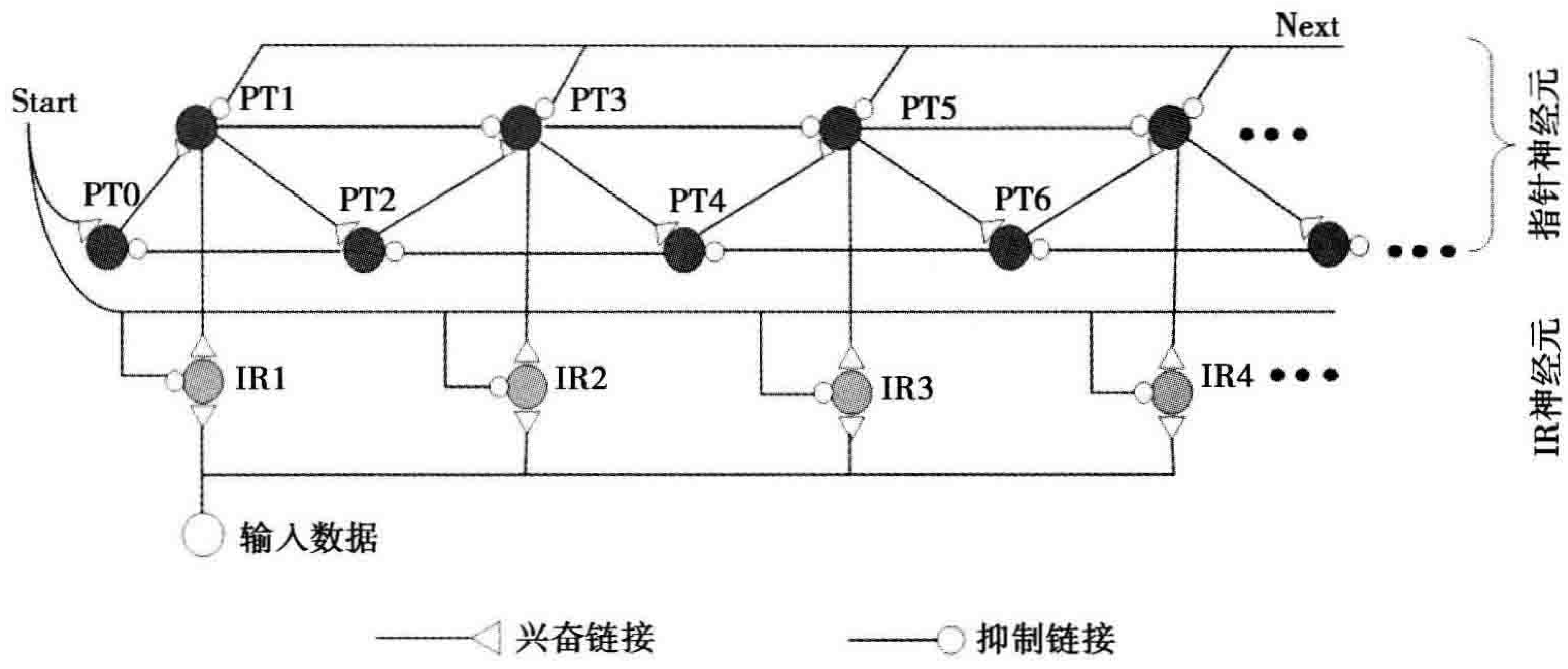


图 7-6 IR 的结构

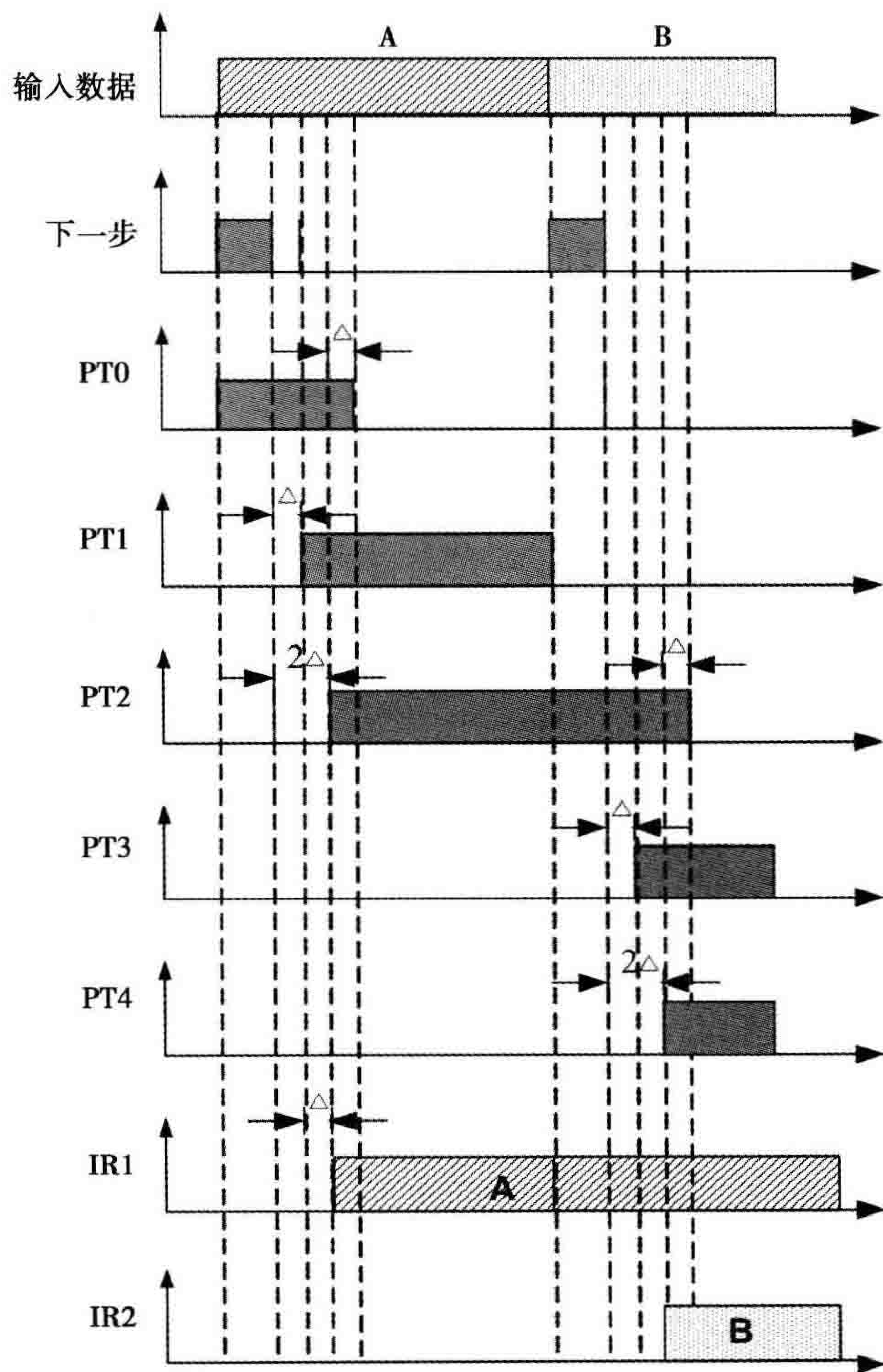


图 7-7 IR 中神经元激活的时控示意

IR1 中。假定一段时间后，出现另一个数据“B”，那么将产生一个“Next”脉冲信号。这个信号抑制了所有较高的指针神经元(upper pointer neuron) PT1、PT3、PT5 等。但是，PT2 保持激活状态。当“Next”信号变低时，经过延迟 Δ 时间后，PT3 因 PT2 的刺激而被激活。和前面一样，PT3 延迟 Δ 时间后，PT4 被激活。同时，因为 PT3 的刺激和输入数据神经元，IR2 被激活。数据“B”将被存储在第 2 个输入寄存器 IR2 中。整个时间图表如图 7-7 所示。

这个过程一直进行到序列中所有的输入数据都出现。在这个方案中，低层指针神经元提供抑制反馈来消除前一指针神经元的刺激，并激活下一指针神经元。这个方法对所学习的序列形成了长时记忆(LTM)。通过一段时间的训练衰减后，才可能使这些记忆改变。

7.5.2 序列预测

1. 预测机制

预测一个输入序列是序列学习的关键组成部分(Starzyk & He，2007)。通过正确的预测，智能系统能够表示它知道这个序列，从而不需要进行额外的学习。如果出现错误，那么就需要修改长时记忆来学习新的序列。

预测的第一阶段是竞争阶段。存储在 LTM 中的几个序列通过竞争来决定可能与输入序列相符合的某一序列。一旦确定了这样一个序列，它会对后续的输入进行预测(值得注意的是，LTM 的一个高层节点被内部进程触发后，如果 LTM 简单地播放一个存储的序列，则会使用相同的机制)。多优胜者检测神经元(MWDN) 是用来检测是否存在多个拥有训练过的链接(trained link)的获胜者。这是通过把 MWDN 的阈值设置为 2 实现的。因此，当输出层有两个或多个获胜者时，它会被激活(当所有获胜者的权值总和相同时会出现这种情况)。通过抑制链接，MWDN 的输出连接到 LFN 和所有的 PN。这为设定适当的学习标记信号提供了一种机制。LFN 的输出通过兴奋链接连接到学习神经元。从输入序列神经元的末尾(ESN)到兴奋链接，整个系统提供了适当的机制来设定学习信号。值得注意的是，每一层都需要 ESN 来指示这一层输入序列的末尾。另外，如果一个 ESN 神经元在高层被激活，那么它将在所有较低层自动产生一个 ESN 信号。考虑这样的例子，在 LTM 的第 1 层存储字母，第 2 层存储单词，第 3 层存储句子。每一个单词的最后一个字母将会触发第 2 层 ESN 神经元的激活。每一个句子的最后一个单词的最后一个字母将会紧随其后引发第 3 层 ESN 的激活，同时，ESN 信号会被

发送至较低层(此处指第 2 层)。

预测机制竞争阶段的所有可能结果可以归纳为 3 种情况。

(1)情况 1

如果在竞争阶段存在单一的拥有训练过的链接的获胜者，这种架构或者会激活 PCN(即预测正确，到目前为止不需要学习)，或者会激活 LFN(即需要对新序列进行一次性学习)。

在这种情况下，MWDN 不会触发，因为它的阈值为 2。因此，PN 和 LFN 不会被抑制。PN 中的每个 PN 都对应于下一较低层的每个输出神经元。通过“时控复用器”一节中所描述的机制，LTM 通过 MUX 在每个时间步激活一个到 PN 的兴奋映射产生作用，该映射表示这个时间步的网络预测。时间指针(TP)会随着 LTM 中出现的每个新标志(模式)增加。从图 7-2 中也可以看出，每个 PN 从 MWDN 接收一个抑制投射，从 MUX 接收兴奋投射。因为在这种情况下 PN 没有被 MWDN 抑制，所以 LTM(通过 MUX 作用)激活与输入序列预测标志相对应的特定 PN。每个预测神经元和与它相对应的下一低层的输出神经元构成 PMN 的两个输入信号。PMN 的激活验证了预测标志与输入标志相符合，如图 7-8 所示。

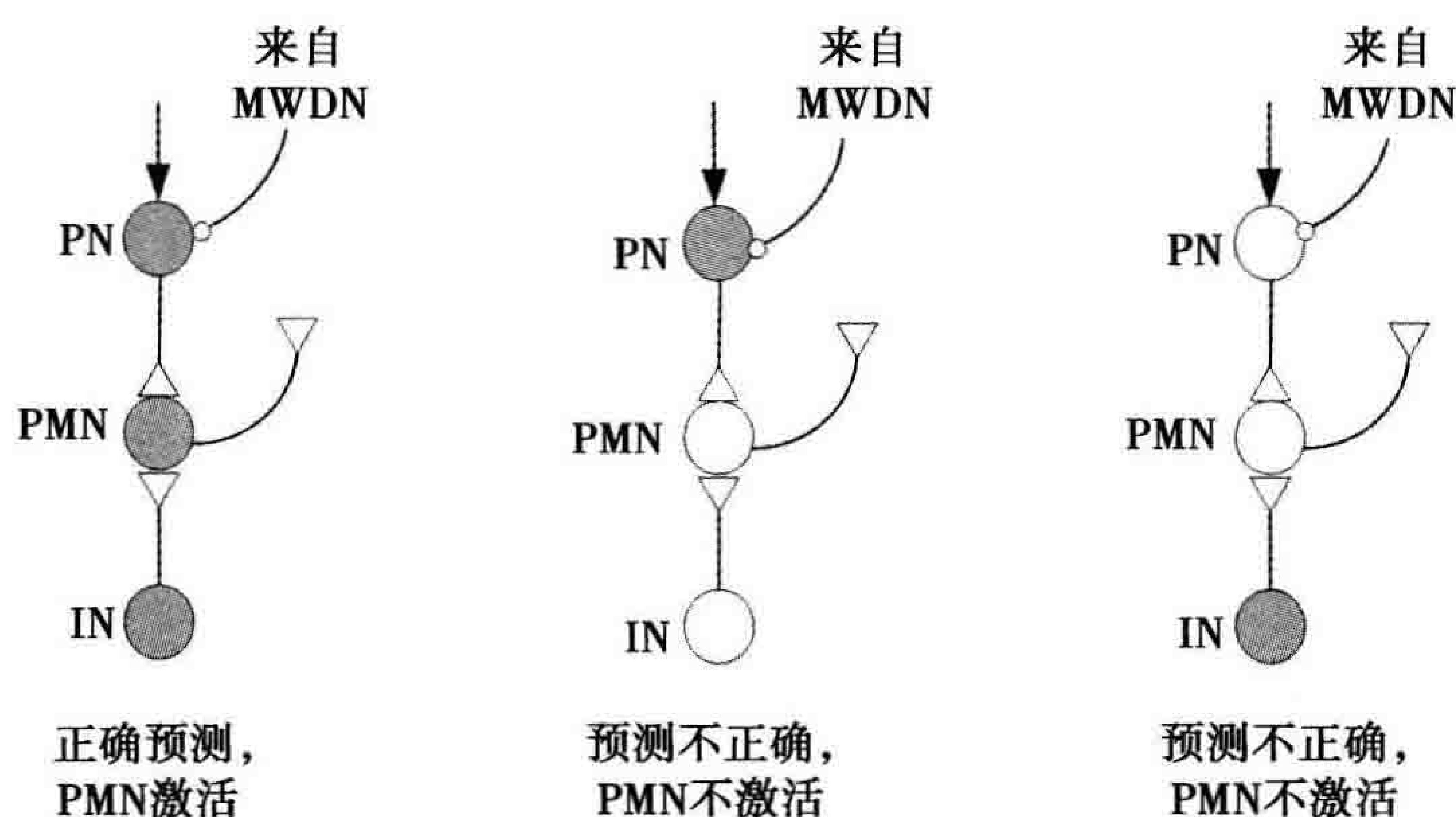


图 7-8 PMN 的激活机制

所有的 PMN 都有输出信号连接到 PCN，如图 7-9 所示，这个神经元如果被激活，就表明这是一个正确的预测。

如果不匹配，那么 LFN 会自动设定(没有来自 MWDN 的 PCN 抑制)。LFN 会继续存在并且序列会继续进行直到 ESN 激活。因此，LFN 和 ESN 的激活触发了学习神经元，如图 7-10a 所示。如果与 PMN 的输出相匹配，则 PCN 被激活，因此 LFN 被抑制。图 7-10b、c 展示了 LN 的另外两个条件。图 7-10b 表明，如果只有 LFN 激活(即预测不正确)，那么 LN 神经元不会被激活，因为在这种情况下 ESN

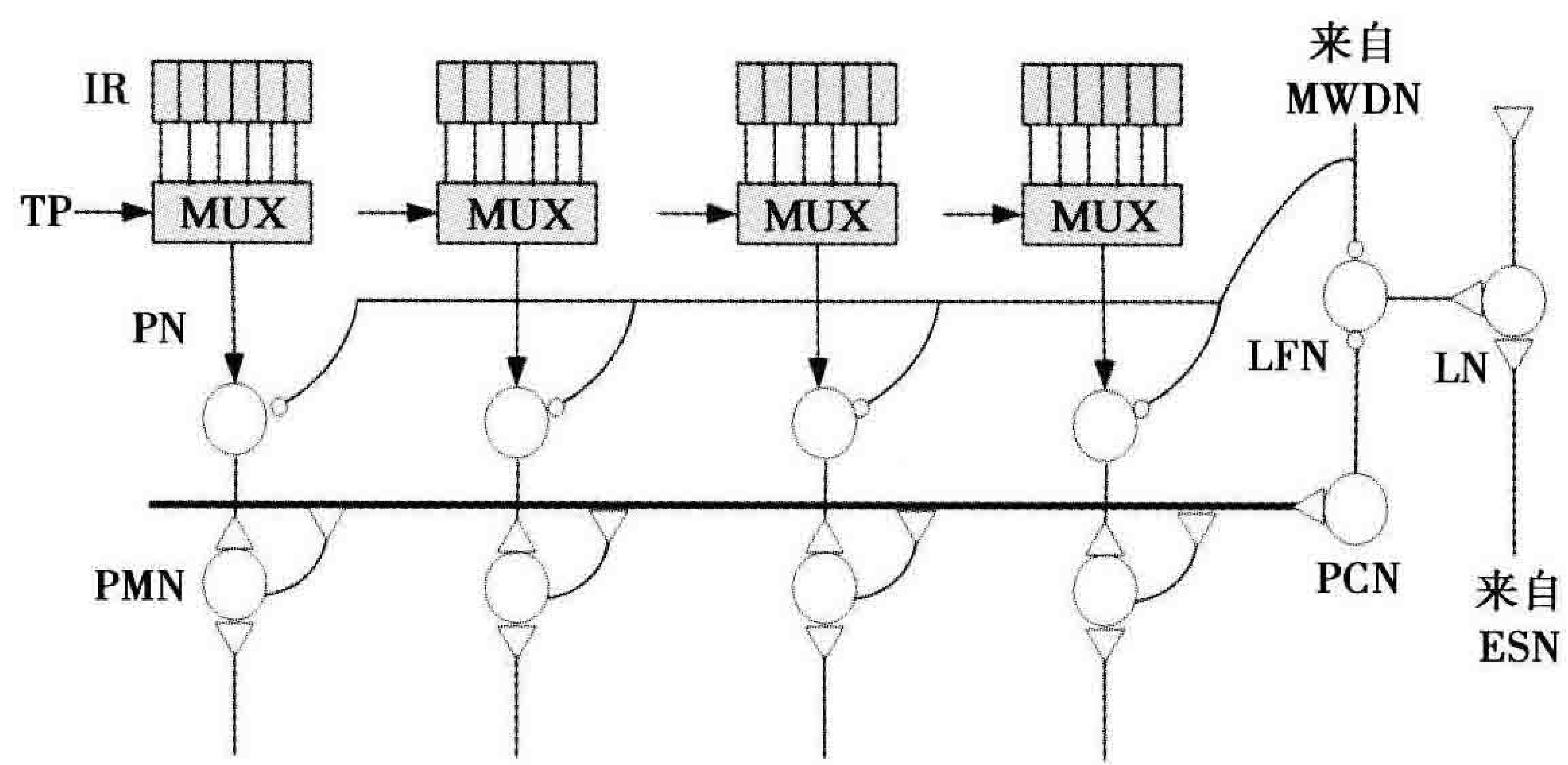


图 7-9 预测机制

没有被激活。图 7-10c 表明，如果只有 ESN 激活，那么 LN 不会被激活，因为 LFN 没有被激活(即预测正确)。

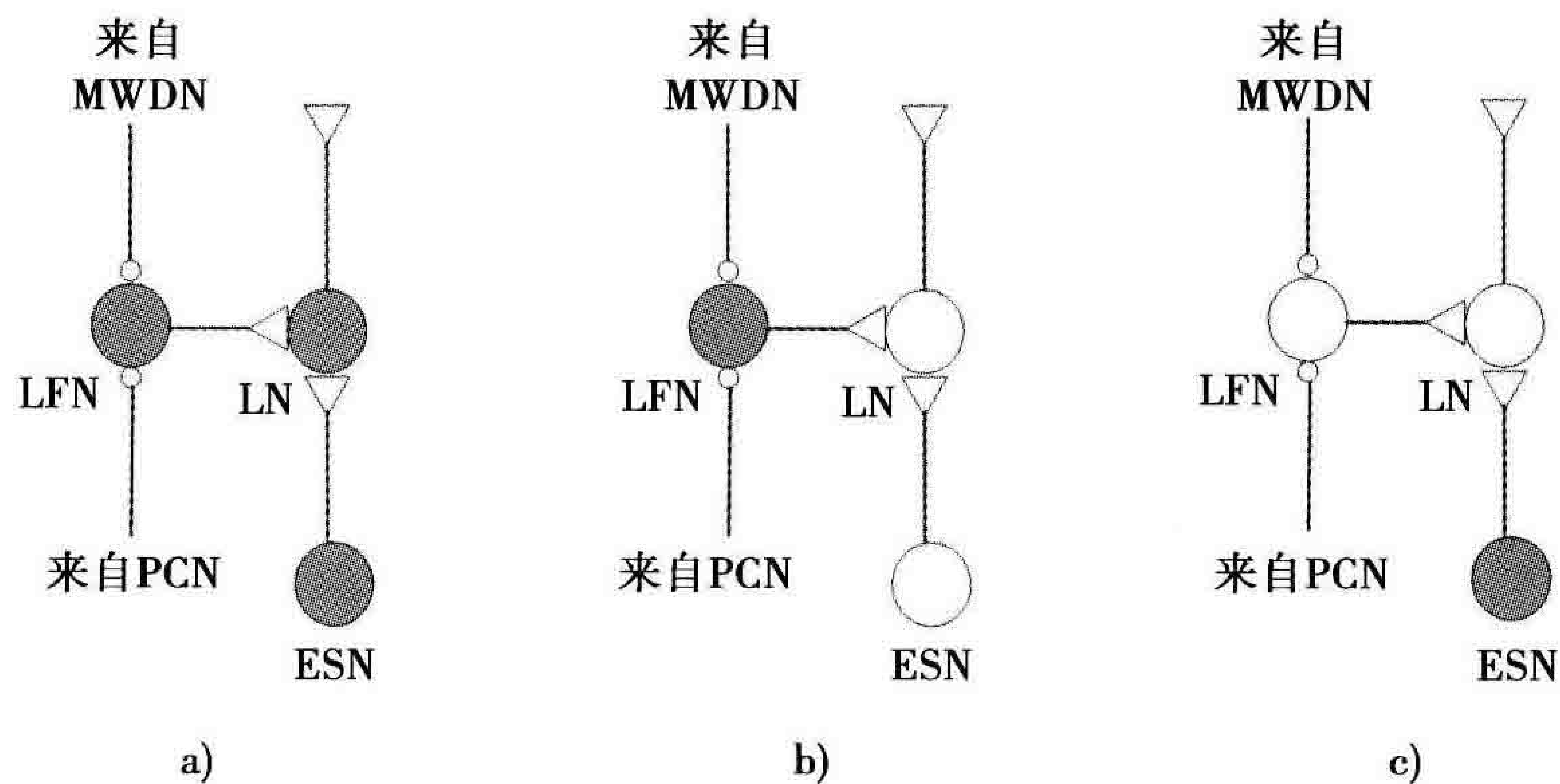


图 7-10 学习神经元的激活机制

(2)情况 2

如果在竞争阶段存在多个拥有训练过的链接的获胜者，则不做预测。

假设有 $n(n\geqslant2)$个输出神经元，并且全都拥有经过训练的、接到特定 IR 神经元的链接。那么，当 IR 神经元被激活后，上述 n 个神经元都会被激活。因为 MWDN 的阈值设定为 2，所以 MWDN 会被激活。从图 7-2 的系统级架构可以看出 MWDN 会抑制 PN 和 LFN，因此，MWDN 会阻止系统预测下一个数据是什么。考虑如图 7-11 所示的情况，假设两个单词“miss”和“mom”已经存储在 LTM 中(图 7-11 中仅显示兴奋性链接)，新序列是“mit”，那么，当序列的第 1 个符号“m”输入时，两个神经元 n1 和 n3 用训练过的链接获胜(权值等于 1)。然后 MWDN 的阈值达到 2 并被激活。如图 7-11 所示，MWDN 抑制 PN 和 LFN。由于两个神经元都是具有训练过的链接的获胜者，所以对网络来说试图进行预测是不成熟的(所有 PN 神经元的抑制

减少了能量消耗)。当第 2 个符号"i"出现时，神经元 n1 获胜，因为它从 IR 接收了两个兴奋性映射，而神经元 n3 接收到了抑制性映射。在这种情况下，只有一个具有训练过的链接的获胜者(即情形 1)。MWDN 没有激活，消除对 PN 和 LFN 的抑制。通过 MUX 对 TP 信号的控制，n1 会预测下一个符号为"s"，即在这个例子中错误的那个。因此 PMN 会被激活(见图 7-8)，PCN 不会被激活。通过这种方式，LFN 会被激活，因为没有来自 PCN 的抑制。在这个时候，因为 ESN 没有被激活，所以 LN 也没有被激活。当第 3 个符号"t"出现时，n1 和 n3 都接受来自 IR 的抑制，并且都不会获胜。不失一般性，假设 n2 是获胜者，这将导致情况 3 中存在不具有训练过链接的单一获胜者。

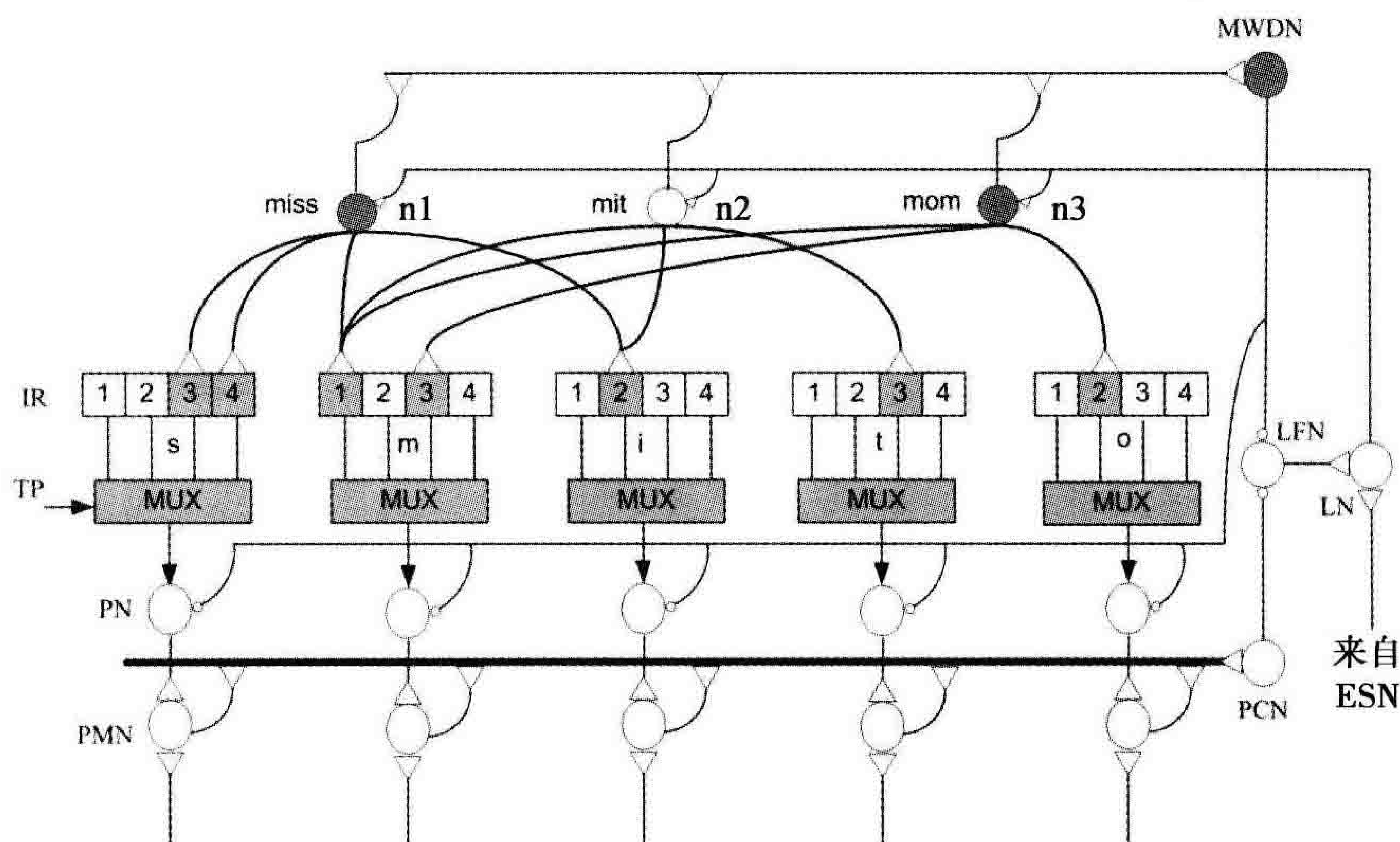


图 7-11 LTM 和多获胜者

(3)情况 3

如果在具有未训练链接的输出神经元中存在单一获胜者，那么在序列的结尾将会发出一个学习信号，对这个序列执行单次学习。

在这种情况下，MWDN 不被激活，因为获胜者具有未训练链接，所以 PN 不被激活。因此，PCN 不被激活，但允许 LFN 激活。LFN 保持活跃状态直到 ESN 激活。然后 LFN 和 ESN 的组合会引起 LN 的激活，学习信号激活单次学习并根据之前描述的学习序列"mit"的规则调整权值大小。图 7-11 说明了在学习后加强的连接。

从上述的 3 种情况中，我们可以看出，在竞争阶段的所有条件下，这个模型或者正确地预测序列，或者在一个新序列的结尾执行单次学习，从而学习这样的序列。在接下来的章节中，我们将详细讨论关于预测神经元激活的预测机制和时控 MUX

的设计。

2. 预测神经元的激活

为了执行序列预测，每个 IR 神经元都与一个对偶 IR 神经元(dual IR neuron)有关。WTA 神经元负责存储序列，通过未经训练的链接连接到对偶 IR 神经元。IR 神经元通过训练过的链接连接到它们的对偶神经元。因此激活一个 IR 神经元会自动地激活它的对偶神经元。当一个序列被存储在 WTA 神经元中时，从 WTA 神经元到对偶 IR 神经元的链接是训练过的(图 7-12 中的粗线所示)，该对偶 IR 神经元是序列中活跃的 IR 神经元对应的对偶 IR 神经元。当先前存储的序列再次被输入时，一部分匹配的序列可能会激活这个序列的 WTA 神经元。这将激活组成整个序列的所有对偶 IR 神经元。结构如图 7-12 所示。这种结构与时控复用器相结合，提供了预测机制。

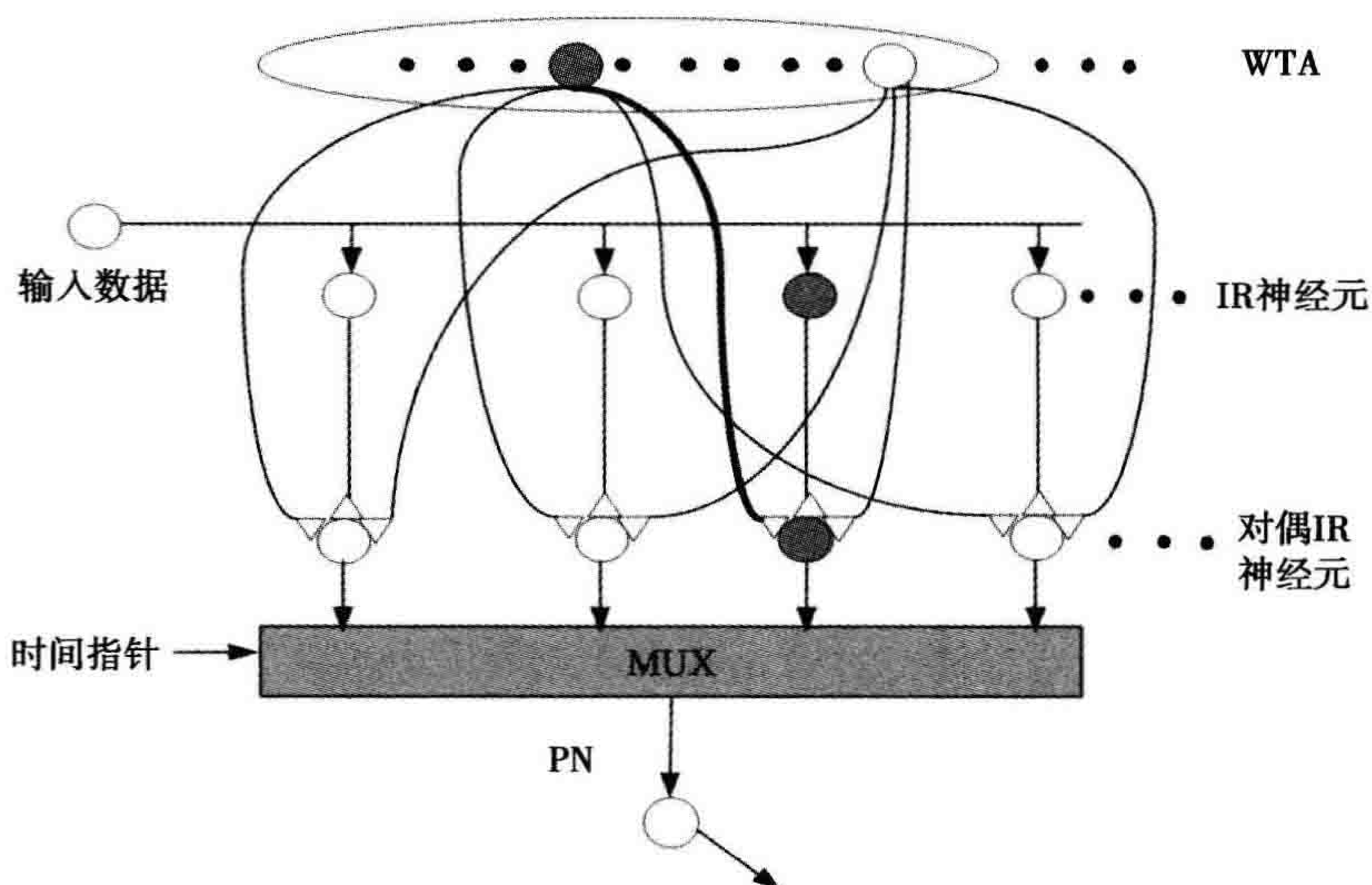


图 7-12 预测神经元的激活

3. 时控复用器

时控复用器的神经网络结构如图 7-13 所示。正如在“预测神经元的激活”小节中所讨论的，从 WTA 的输出激活对偶 IR 神经元，该对偶 IR 神经元表示对每个时间步的预测。在一个由活跃指针神经元给定的时间步内，该对偶神经元激活对应的 IR 输出神经元，随后激活存储序列中下一个元素相应的预测神经元。该预测数据与 PMN 连接，该 PMN 与具有实际数据的 PN 比较，如果预测正确，则被激活。

假设经过 WTA 竞争，一个存储“miss”的输出神经元是激活的，那么，该神经元将激活它的对偶 IR 神经元。当时间指针增加到 3 的时候，较高的指针神经元 PT5 激活。在这种情况下，活跃的 PT5 神经元和对偶 IR 神经元(存储数据“s”)会同时激

活对应的时控 MUX 神经元(在图 7-13 中活跃的神经元用灰色圆圈表示)。这反过来会激活对应的 PN，并且，这个激活信号会被发送到 PMN，以用来检测与预测数据的匹配或误匹配。应该注意的是：图 7-13 中仅说明了存储字母“s”的 IR 结构，而存储“m”和“i”的 IR 结构没有在图 7-13 中说明。

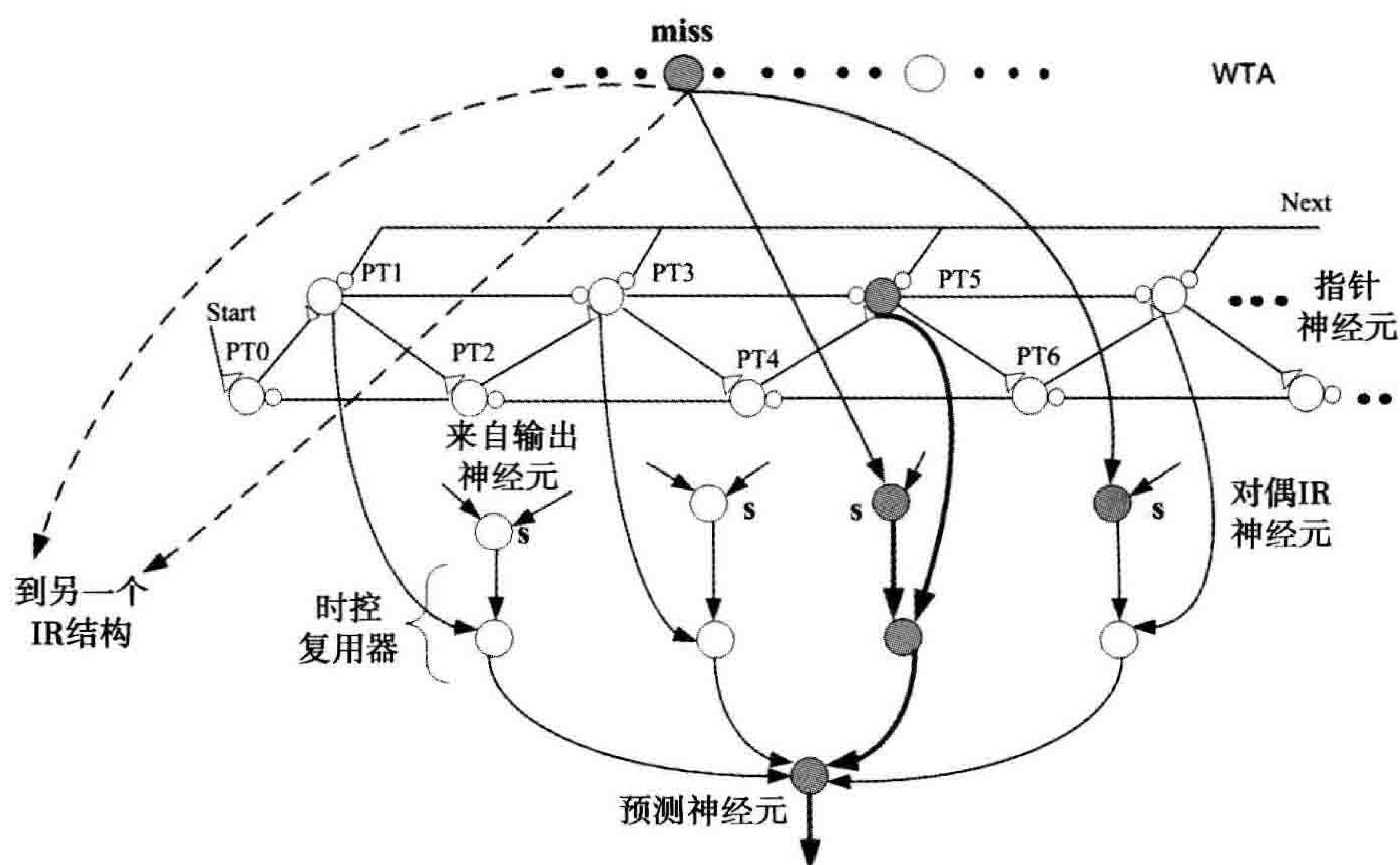


图 7-13 时控复用器

7.5.3 序列检索

储存的时-空序列可以在任意层被活跃神经元检索，这种检索是通过内部过程或者通过与第 1 层(感觉)输入信号相关联实现的。序列检索过程中的一个关键因素是序列中每个元素的持续时间。通过缩放这个时间，可以实现不同的检索速度。Wang 和 Arbib(1993)提出了一种用来存储检索时间间隔的机制，其中检索时间与输入序列的测定时间密切相关。在实际应用中，尽管与原始序列同步存储序列很重要，但一般情况下没有必要。例如，如果赋予一个人一项需要一系列操作的复杂任务，序列的执行时间是由完成每项任务所需要的时间决定的，并且这个执行时间可能是事先未知的或者依赖于任务被执行的情况。生物体通常依赖于感觉运动协调来提供执行存储序列的适当时间。检索的速度取决于自然延迟和接收到感觉输入反馈的时间，其中，输入反馈证明了对序列中特定元素的检索已经完成，序列元素检索的完成包括对存储序列的下一个元素的表述。因此，对时间序列的检索过程是自计时的，不依赖于任何内部时钟，而依赖于与环境的交互作用。这可能是许多需要时空存储器的实际应用所期望的特性。

7.6 内存需求

本节简要讨论上述模型的内存需求。考虑图 7-2 中的分层结构的第 1 层，假设该结构具有 m_1 个来自较低层(如图 7-2 中的 0 层)的输入神经元，那么，PN 和 PMN 的数量均为 m_1。假设在这一层，输出 WTA 神经元(为下一更高层提供输入)的数量为 m_2，并且这一层可以存储的最长序列的长度为 l。在这种情况下，IR 神经元、对偶 IR 神经元和复用器神经元的总数量等于 $m_1 \times l$。由于 PT 神经元可以被所有的 IR 神经元共享，因此所需的 PT 神经元的总数量为 $2l$。因此，在这个特定的分层结构中，所需的神经元的总数量为 $2m_1+m_2+3m_1\times l+2l$。

现在估计所需的互连链接的数量。从图 7-2 可以看出，WTA 神经元的输入与占主导地位的对偶 IR 神经元之间的链接的总数量等于 $m_1\times m_2\times l$。IR 结构内 PT 神经元之间的链接数为 $4l-3$(见图 7-6)。其他的链接包括 PMN 与前一层的输出神经元(m_1)之间的链接，PMN 与 PN(m_1)、较高的指针神经元与 MUX 神经元($l\times m_1$)、MUX 与 PN($l\times m_1$)之间的链接。因此，所需的链接总数量为$(m_1\times m_2\times l)+(4l-3)+(2+2l)\times m_1$。

例 7.1 模型的内存需求

考虑字母-单词-句子的分层结构，假设要在第 2 层存储 10 000 个单词，在第 3 层存储 1000 个句子(见图 7-2)，并且最长的单词包含 10 个字母，最长的句子包含 20 个单词。因此第 2 层和第 3 层所需的神经元的总数量分别为 1.08×10^4 和 6.21×10^5，第 2 层和第 3 层所需的链接数量分别为 2.6×10^6 和 2.0×10^8。这个估算粗略地给出了该模型的内存需求。

7.7 多序列的学习和预测

到目前为止，已经讨论了所提出的模型如何能够准确有效地实现对复杂序列的存储、预测和检索。下面给出一个完整的例子来说明这个模型是如何学习和预测多重序列的。

为了强调序列学习机制，假设来自环境的每个传感输入都会激活第 1 层输出端相对应的获胜神经元。因此，我们将重点放在图 7-2 中的第 1 层上。正如 Wang 和 Yuwono(1995)所描述的，令“＃”为层级序列结束标志符。假设一个需要存储和检索的多重序列为：“mis＃mit＃miss＃mit＃”，如图 7-14 所示。不失一般性，假设这个层次结构具有 3 个输出神经元、27 个输入神经元(表示完整的字母表加上结束

标志符“#”)。每个输出神经元完全连接到输入寄存器的所有输入神经元上，并且神经元的突触的初始权值被设置为 $0.001<w_1<0.01$。

当第一个序列的第一个符号“m”被激活时，时间指针被设置为1。因为所有权值的初始值都是随机设定的，不失一般性，假设神经元1(n1)为获胜者。由于之前没有训练，所以现在也没有预测。因此 PCN 没有激活，但这引起了 LFN 被激活。LFN 的激活状态一直保持到第一个序列结束时 ESN 被激活。LFN 和 ESN 联合激活 LN，并发出学习信号，触发单次学习，获胜者权值调整为与7.5.1节中的一样(即兴奋性权值被设置为1，抑制性权值被设置为－100)。

当第二个序列的第一个符号“m”被激活时，TP 被设置为1。以前被激活的神经元(n1)成为单一获胜者，因为它接收了来自 IR 初始位置的所有兴奋性映射。n1 通过 TP 信号的多重控制预测下一个符号为“i”。在这种情况下，预测是正确的，相应的 PMN 激活并激活 PCN，抑制 LFN。当第二个符号“i”出现在模型中，TP 增加到2。n1 依然是唯一的获胜者，因为它具有来自相应 IR 神经元前两个位置的两个兴奋性链接，并且没有抑制性链接。相应的 PN 神经元预测“s”为下一个输入符号。因为这个预测是不正确的，所以没有一个 PMN 激活，而且 PCN 也没有激活。相应地，LFN 没有抑制和激活。当第三个符号“t”被激活，TP 增加到3。n1 不是获胜者，因为它具有一个来自 IR 的抑制性映射。不失一般性，假设 n2 是获胜者。当序列结束标志符“#”被激活，ESN 激活。当 LFN 和 ESN 都激活时，LN 激活，并发出学习信号，触发单次学习，并适当调整获胜者(n2)的权值。

当第三个单词的第一个符号“m”出现在模型中时，将会有两个获胜者(n1 和 n2)具有训练过的链接，因此，MWDN 会激活。MWDN 的激活会抑制所有 PN 和 LFN 的激活。当第二个符号“i”出现时，MWDN 再次激活。当第三个符号“s”被激活时，n1 是单一获胜者。MWDN 不激活，并且 PN 和 LFN 的抑制被解除。n1 预测的下一个符号是“#”，这是错误的。当第四个符号“s”被激活，n1 和 n2 均被抑制。不失一般性，假设 n3 是获胜者。因为 n3 没有训练过的链接，所以不执行预测。因此，PCN 不激活，LFN 激活。这个过程一直持续到“#”被激活，并且 ESN 激活。LFN 和 ESN 的激活联合发出学习信号，执行单次学习，并调整 n3 的权值。

当第四个序列的第一个符号“m”被激活时，则有3个获胜者(n1、n2 和 n3)。MWDN 激活，并抑制 PN 和 LFN。当第二个符号“i”被激活时，这3个神经元仍然是获胜者。MWDN 激活，并抑制 PN 和 LFN。当第三个符号“t”被激活时，n2 是拥有训练过的链接的单一获胜者。MWDN 没有激活，并且 n2 会正确地预测下一个符

号“#”。当序列的最后一个符号“#”被激活时，ESN激活。因为在预测正确的时候LFN没有激活，所以LN没有激活。因此，对最后一个序列不需要进行学习。

图7-14表示了上述实例中神经元激活活动的全过程。该模型在3个输出神经元

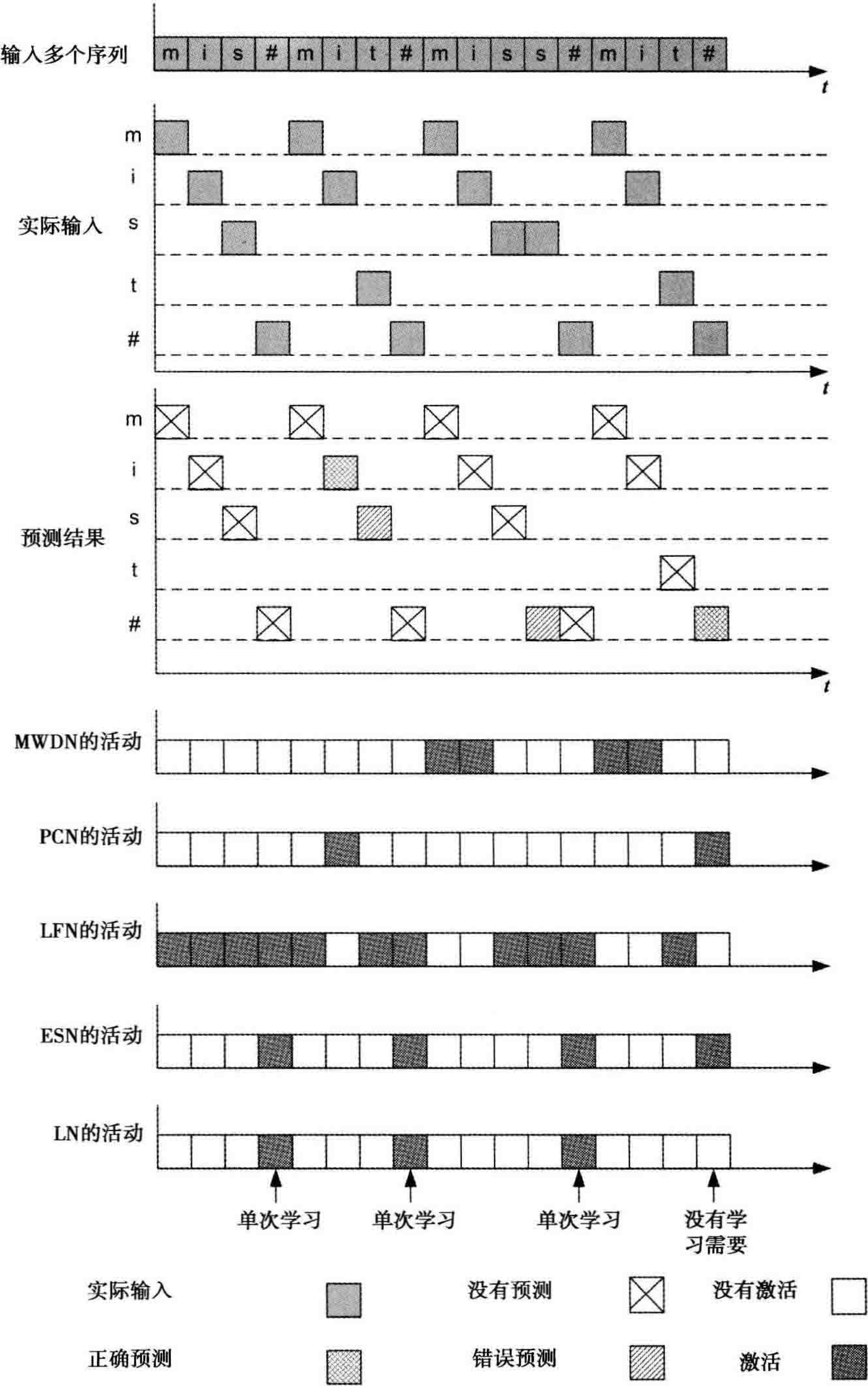


图7-14 多序列的学习和预测

中存储了4个序列(第二个和最后一个序列存储在相同的神经元n2中)。大家可能注意到该基于模型的预测没有将学习和检索过程分离开，这使得这个模型对序列的学习和检索更有效。

7.8 案例研究

本节模拟了一个4层的分层结构，具有字母、单词、句子和流行歌曲“This land is your land”的一节，以说明所提出的分层序列学习模型(Starzyk & He，2007)。

从环境中接收的原始感觉输入数据为20×20像素的扫描图像，包含完整的字母表和3个特殊字符：空格、点、分号，其中3个特殊字符分别是词级、句子级和小节级输入序列的结尾。在学习和回放之间没有区别，意味着对于每个输入序列，该模型或者正确预测这个序列，或者在某一层中的序列结尾处进行单次学习。图7-15给出了该模型的模拟结果。

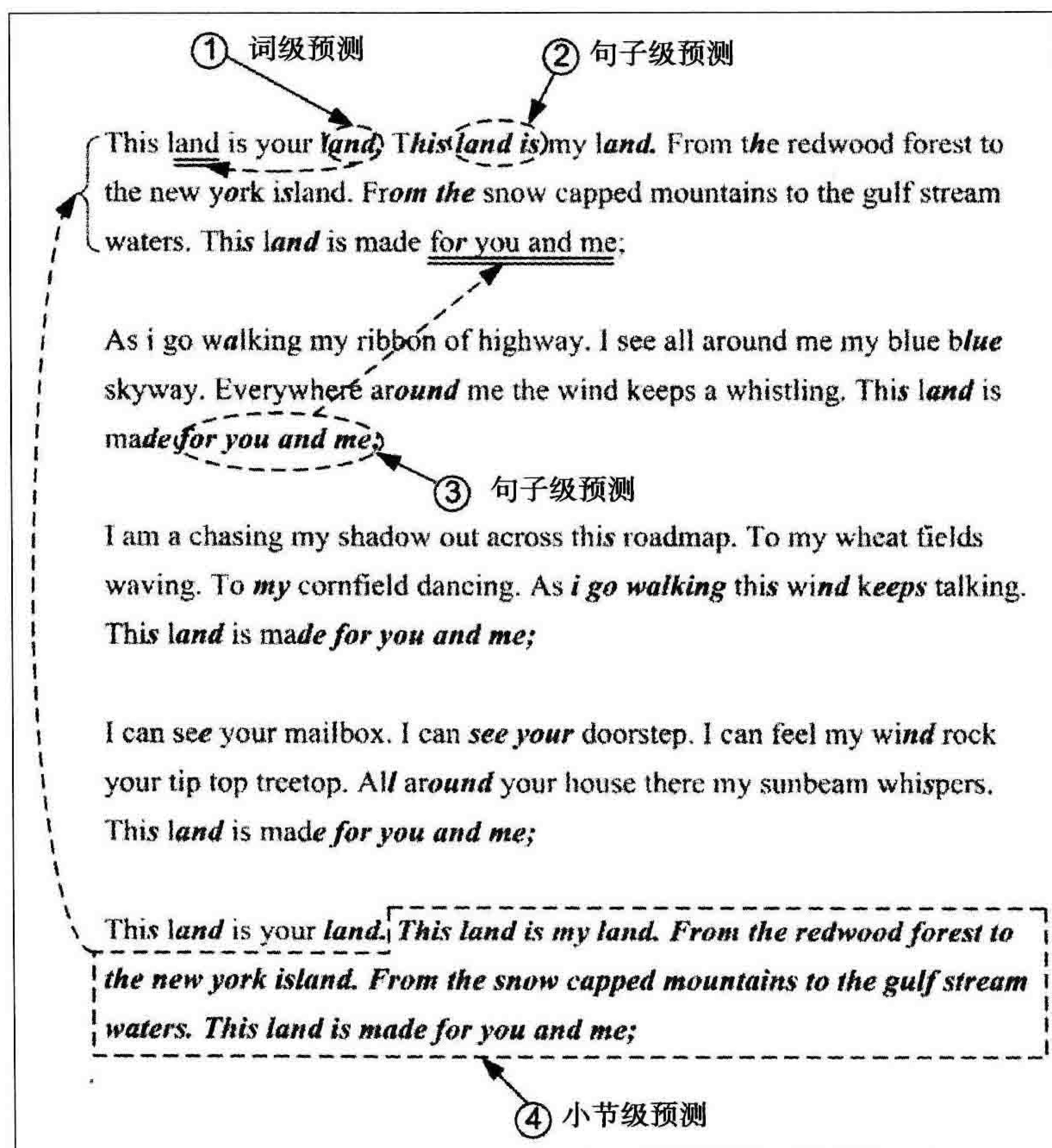


图7-15 时间序列学习模型的仿真结果

浅色字体的文字是输入序列，粗斜体文字代表正确预测的序列元素。①表示在单词级的正确预测。例如，当第二个“*land*”的第一个字母“*l*”被激活时，模型正确地预测到下一个符号“*a*”，因为它已经在“*land*”第一次出现时学习了序列“land”。图 7-15 中的②和③表示句子级正确的预测。“*This land is made for you and me*”在第二节的结尾部分重复，因此当“*This land is made*”出现时，正确预测到词语“*for you and me*”。“*land is made for you and me*”不能在“*This*”之后被正确预测的原因是还存在其他的句子，例如：“*This land is your land*”、“*This land is my land*”，它们的前三个词是相同的。因此，“*This*”出现之后 MWDN 神经元激活，从而抑制预测神经元的激活。图 7-15 中的④表示小节级的正确预测。最后一个小节是第一个小节的重复。因此，在小节的第一个句子之后，模型正确地预测到小节的其余部分。

该模型在存储这首歌之后，应该已经具有根据一个输入提示预测这首歌的能力，或者在不破坏已存储序列的前提下学习一个新的序列。例如，如果给系统一个提示“e”，那么在单词级有一个独一无二的神经元存储单词“everywhere”。因此，模型会输出词“everywhere”。这个神经元在词级激活后，会触发句子级神经元的激活。在这种情况下，句子级的独一无二的神经元是获胜者，该神经元会重现从句子级到单词级的序列，最终生成原始序列。图 7-16 显示了字符“e”作为模型的提示被激活后的仿真结果。

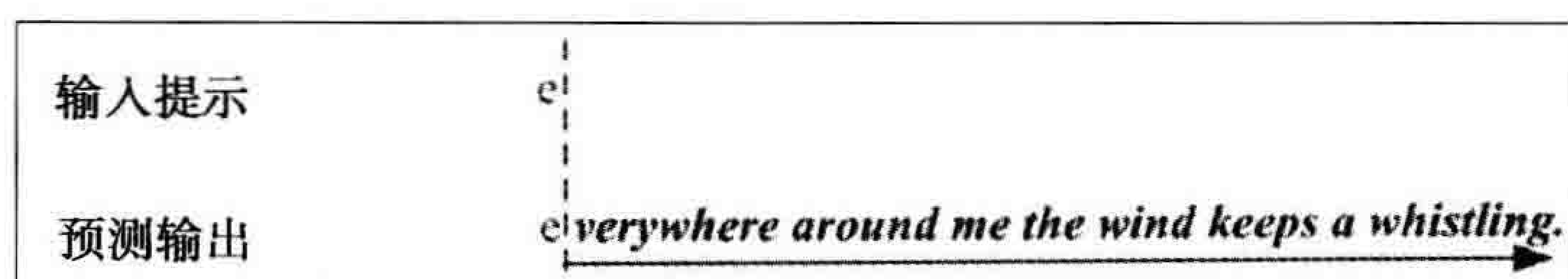


图 7-16 基于输入提示的预测结果

此外，如果模型中出现一个新的单词、句子、小节，它能够在不破坏先前存储序列的前提下进一步学习新序列，并且该学习序列可以被用于联想预测。

7.9 总结

本章的要点包括：

- 本章提出的模型具有分层结构、预测机制和增量学习的特性。在提出的模型中，用一种改进的 Hebbian 学习机制识别模型最低层的输入模式，在每个层级中，用胜者为王机制为其上一层的输入选择神经元。

- 本章提出的模型能够有效地处理大规模、多元化、复杂序列的学习、存储和检索。分层结构和预测机制是该模型的主要组成部分，它能够使模型在每一层积极地预测下一个输入。只要预测到的输入被正确验证，那么在该层的任意层级上都不需要再学习。当检测到某层的某一给定层级出现误匹配时，这一层和所有更高层级上的新序列都需要通过单次学习机制再次学习。
- 本章的研究结果表明，序列学习模型是智能系统的重要组成部分。本章提出的模型是面向硬件的，并且对构建工程设备具有重要意义。在这个模型中，WTA 的使用和对单一神经元的符号表示，挑战了业界广泛持有的观点(即生物大脑使用分布式表示)。此外，该模型使用连续事件的类记忆存储，这不同于在许多研究中使用的移位寄存器结构。在提出的模型中，当序列中的新元素在系统中出现时，每个新元素被存储在 IR 中的一个指定位置，而不是移动所有输入。
- 提出的模型采用分层的方法通过分块机制延伸学习的概念，只需要输入数据的单一表示来学习整个序列。分层结构的任意层级的任意序列通过自组织分配必需的资源，从而学习。该方法能够存储任意的复杂序列，只要不同子序列的数量小于不同层级上神经元总数量表示的存储容量。我们认为这种结构模型是一种更加自然的序列学习方法，并且在学习事件的上下文范围内允许序列自然分组。
- 除了本章列举的应用问题，这种分层序列学习模型还在不同领域具有广阔的应用前景。例如，该模型可以被用于智能文字处理或语音处理。模型可以学习一个人的讲话方式，然后根据一些提示识别并预测他或她的讲话内容。也许，序列学习最重要的应用可能会在具身智能系统中，这需要对基于事件或行为的观察序列所增加的信息进行不断的预测。

参考文献

Anderson, J. (1995). *Learning and memory*. New York: Wiley.

Ara'ujo, A. F. R., & Barreto, G. A. (2002). Context in temporal sequence processing: A self-organizing approach and its application to robotics. *IEEE Trans. Neural Networks*, *13*(1), 45–57.

Bose, J., Furber, S. B., & Shapiro, J. L. (2005). An associative memory for the on-line recognition and prediction of temporal sequences. *Proc. Int. Joint Conf. Neural Netw.*, pp. 1223–1228.

Bower, G. H., Thompson-Schill, S., & Tulving, E. (1994). Reducing retroactive interference: an interference analysis. *J. Experimental Psychology: Learning, Memory, and Cognition*, 51-66.

Chartier, S., & Boukadoum, M. (2006). A sequential dynamic heteroassociative memory for multistep pattern recognition and one-to-many association. *IEEE Trans. Neural Netw.*, *17*(1), 59-68.

Chun, M. M., & Jiang, Y. (1998). Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cognitive Psychology*, *36*, 28-71.

Elman, J. L. (1990). Finding structure in time. *Cognitive Sci.*, *14*, 179-221.

Farkas, I., & Li, P. (2002). DevLex: A self-organizing neural network model of the development of lexicon. In W. D. Grey C. D. Schunn (Eds.), *Proc. Int. Conf. Neural Information Processing*. Singapore: Nanyang Technology University.

George, D., & Hawkins, J. (2005). Invariant pattern recognition using bayesian inference on hierarchical sequences *[Online], Available: http://www.stanford.edu/dil/RNI/DilJeff TechReport.pdf*.

Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: Times Books.

Hawkins, J., & Blakeslee, S. (2007). Why can't a computer be more like a brain. *IEEE Spectrum*, *44*(4), 20-26.

Hawkins, J., & George, D. (2006). Hierarchical temporal memory-concepts, theory, and terminology. *Numenta Inc. [Online], Available: http://www.numenta.com/*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735-1780.

Jacobsson, H. (2005). Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation*, *17*(6), 1223-1263.

James, D. L., & Miikkulainen, R. (1995). STARDNET: A self-organizing feature map for sequences. In G. Tesauro, D. S. Touretzky T. K. Leen (Eds.), *Advances in neural information processing systems* (Vol. 7).

Jiang, Y., Olson, I. R., & Chun, M. M. (2000). Organization of visual-short term memory. *J. Experimental Psychology: Learning, Memory and Cognition*, *26*, 683-702.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proc. Conf. Cognitive Sci. Soc.*, pp. 531-546.

Kumar, A., & Jiang, Y. (2005). Visual short-term memory for sequential arrays. *Memory and Cognition*, *5*(7), 650-658.

Lewkowicz, D. J. (2004). Preception of serial order in infants. *Development Science*, *7*(2), 175-184.

Lewkowicz, D. J. (2006). Learning and discrimination of audiovisual events in human infants: The hierarchical relation between intersensory temporal synchrony and rhythmic pattern cues. *Development Science*, *39*(5), 795-804.

Lewkowicz, D. J., & Ghazanfar, A. A. (2006). The decline of cross-species intersensory perception in human infants. *Proc. National Academy of Sciences*, 103, 6771-6774.

Lewkowicz, D. J., & Marcovitch, S. (2006). Preception of audiovisual rhythm and its invariance in 4- to 10-month-old infants. *Development Science*, *48*, 288-230.

Li, P., & Farkas, I. (2002a). Bilingual sentence processing. In R. Heredial J. Altarriba (Eds.), (pp. 59-85). North Holland: Elsevier Science.

Li, P., & Farkas, I. (2002b). Modeling the development of lexicon with devlex: A self-organizing neural network of lexical acquisition. In W. D. Grey C. D. Schunn (Eds.), *Proc. Annual Conf. Cognitive Science Society*. Mahwah, NJ: Erlbaum.

Li, P., & Farkas, I. (2002c). Modeling the development of lexicon with devlex: A self-organizing neural network of lexical acquisition. *Proc. Annual Conf. Cognitive Science*

Society.

Liu, K., & Jiang, Y. (2005). Visual working memory for briefly presented scenes. *J. Vision*, *5*(7), 650–658.

Manning, C. G. N., & Witten, I. H. (1998). Identifying hierarchical structure in sequences: a linear-time algorithm. *Cognitive Psychology*, *36*, 28–71.

McClelland, J. L., & Elman, J. L. (1986a). The trace model for speech preception. *Cognitive Psychology*, *18*, 1–86.

McClelland, J. L., & Elman, J. L. (1986b). Interactive process in speech preception: The TRACE model. In *Parallel distributed processing explorations in the microstructure of cognition. Vol. 2, Psychological and biological models* (pp. 58–121).

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter preception: Part i – an account of basic findings. *Psychological Review*, *88*, 375–407.

Miikkiulainen, R. (1993). *Subsymbolic: Natural language processing: An integrated model of scripts, lexicon, and memory*. Cambridge, MA: MIT Press.

Miikkulainen, R. (1990). *DISCERN: A distributed artificial neural network model of sScript processing and memory*. Unpublished doctoral dissertation. (Technical Report UCLA-AI-90-05)

Miikkulainen, R. (1992). Trace feature map: A model of episodic associative memory. *Biological Cybernetics*, *66*, 273–282.

Moriarty, D. E., & Miikkulainen, R. (1999). *Advances in the evolutionary synthesis of neural systems*. Cambridge, MA: MIT Press.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

Pollack, J. B. (1991). The induction of dynamical recognizers. *Machine Learning*, *7*, 227–252.

Schneider, D. W., & Logan, G. D. (2006). Hierarchical control of cognitive processes: Switching tasks in sequences. *J. Experimental Psychology*, *135*(4), 623–640.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed developmental model of word recognition and naming. *Psychological Review*, *96*, 523–568.

Starzyk, J. A., & He, H. (2007). Anticipation-based temporal sequences learning in hierarchical strcuture. *IEEE Trans. Neural Networks*, *18*, 344–358.

Sun, R., & Giles, C. L. (2001). Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intell. Syst.*, *16*(4), 67–70.

Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction process. *Neural Networks*, *16*, 11–23.

Tani, J., & Nolfi, S. (1999). Learning to preceive the world as articulated: An approach for hierarchical learning in a sensory-motor systems. *Neural Networks*, *12*, 1131–1141.

Wang, D., & Arbib, M. A. (1990). Complex temporal sequence learning based on short-term memory. *Proc. IEEE*, 78, 1536–1543.

Wang, D., & Arbib, M. A. (1993). Timing and chunking in processing temporal order. *IEEE Trans. Systems, Man, and Cybernetics*, *23*(4), 993–1009.

Wang, D., & Yuwono, B. (1995). Anticipation-based temporal pattern generation. *IEEE Trans. Systems, Man, and Crbernetics*, *25*(4), 615–628.

Wang, D., & Yuwono, B. (1996). Incremental learning of complex temporal patterns. *IEEE Trans. Neural Networks*, *7*(6), 1465–1481.

Wang, L. (1998). Learning and retrieving spatio-temporal sequences with any static associative neural network. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process*, *45*(6), 729–739.

Wang, L. (1999). Multi-associative neural networks and their applications to learning and retrieving complex spatio-temporal sequences. *IEEE Trans. Syst., Man, Cybern B, Cybern*, *29*(1), 73–82.

Zhao, X., & Li, P. (2007). Bilingual lexical representation in a self-organizing neural network. In D. S. McNamara J. G. Trafton (Eds.), *Proc. Annual Conf. Cognitive Science Society*.

第 8 章 机器智能的硬件设计

最终建议

本书最后一章对自适应智能系统硬件设计提出了一些建议。虽然现有的许多研究工作集中在机器智能的软件开发上，但是近期深亚微米电子学和纳米电子学的发展为设计大规模、并行、可扩展、集成化的智能系统硬件平台提供了技术平台。

一般来说，智能系统模型可以在软件环境下模拟，或用硬件，如 VLSI 系统和 FPGA 来构建。与硬件设计相比，软件实现可能更容易一些，但其有自身的固有限制。基于目前的计算机技术，用软件模拟超过 10 000 个神经元的网络是不现实的，这种限制来自有限的计算机运行速度和动态内存。因此，尽管软件系统可以用来测试机器学习模型，但还不足以建立高度集成化的复杂智能系统。随计算机产业的不断发展，未来计算机的效率会提高，可以期望在接下来的 7 年时间内，计算机速度提高 10 倍且随机存取存储器(RAM)扩大 20 倍(Arden 等，2005)(Moore，2009)，这将会使仿真网络的规模增长到成千上万神经元的水平。然而，工业界预期未来将生产的晶体管数量是目前的 60 倍(Moore，2009)，同时，系统级容量将预计增长 60 倍。因此，预计的硬件容量明显比软件速度增长的快。另外，硬件运行的速度将会增加(与计算机速度相同的增长率)，因此，在接下来的 7 年，与软件速度 10 倍的增长率相比，并行硬件的计算能力将增加 600 倍。在将来，模拟机器智能系统神经元网络的软件能力与实现它们的硬件能力之间的差距会变得更大。因此，为实现真正的智能系统，我们需要开发集成化、并行、可扩展的硬件系统，并在多阵列处理器上测试它们的性能。

功率损耗是这种复杂系统的硬件实现的最关键问题之一。今天，一个以 3GHz 的速度运行的简单处理器，消耗的功率大约是 100W，用这样的处理器构建的一个大阵列将会因为成本太高而无法操作。与之相比，拥有10^{11}个神经元的人类大脑消耗的功率大约是 10W。如果我们认定神经元就是智能系统的基本处理器，这相当于

每个神经元消耗10^{-10}W 功率。与今天计算机消耗的 100W 相比较，每个神经元平均少消耗的能量多至 10^{12}W。因此，功率损耗是类脑智能系统硬件设计的关键问题。

观察 N 个数字处理器同时工作的动态能量损耗，可以估算出，能量损耗与 NCV^2f 成比例，其中 C 是单元处理器的活动开关总电容，V 是电源电压，f 是工作频率。假设一个单元处理器的有效开关电容大约为 10nF(这个数字与一般处理器相符合)。这个电容在未来会随着配线幅度的减少而直线下降。由于配线幅度以每 7 年 2/5 的速率在减少(大约是每 3 年 30%的减少量)(Arden 等，2005；Marly，1996)，因此这里没有考虑太多由此引起的预期减少。电压大约是 1V，并且在以后的 7 年不会有明显的减少。因此，如果我们想要设计一个拥有10^{11}个处理器的系统(不承担成本)，其总动能为：

$$P = 10^{11} \times (10 \times 10^{-9}) \times 1^2 \times f$$

要想达到与人脑相同的功率损耗，这样的系统必须以 $f=10^{-2}$ Hz 的频率运行。因为神经元的响应时间估计大约为几毫秒的数量级(这里假设为 5ms)，在相同的功率预算下操作，这样的系统的运行速度只是人脑的$\frac{1}{20\ 000}$。即使假设人脑在任意时间只有 5%的神经元是活跃的，系统操作的运行速度仍然会是人脑的$\frac{1}{1000}$。因此，在现代 VLSI 技术中，大脑级系统唯一可行的实现大概是模拟 VLSI 的实现。例如，为实现神经网络，Mead(1989a，1989b，Mead & Ismael，1989)和 Vittoz(1985，1990a，1990b，1996，1998)中的关于模拟 VLSI 电路的许多开拓性的研究工作为此提供了重要的思想和设计策略。一般来说，模拟处理器需要小两个数量级(因此负载的电容需要是之前的$\frac{1}{100}$)(Bayraktaroglu，Balkir，& Dundar，1996；Bayraktaroglu，Ogrenci，Dundar，Balkir，& Alpaydin，1997)，才可以在较低的电压下运行。在神经元(在真实的神经网络中) 之间进行信息转换的电压电平大约降到 40mV，功率需求才会降到$\frac{1}{600}$。这种小电容负载的组合会将功率损耗降到相当或者低于活动大脑的能耗，从而使具有人脑智能水平的硬件实现从能量的角度可行。

即使解决了功率损耗问题，也无法保证在不久的将来就能建立具有人脑能力的智能系统。例如，现有的 FPGA 芯片能够集成大约 400PicoBlaze 的控制器，这决不需要再设计一个拥有 10^{11}个处理器的系统。即使一个 FPGA 芯片阵列被设计为拥有 10 000 个芯片，能够仿真 4×10^6 个处理器，但随后 7 年的硬件发展能够让具有 10 000 个芯片的系统的处理器数量增加到 10^8 个(假设当前的晶体管数量的增长速度

为每个芯片每18个月增长一倍)。即使具有这样的能力,并行实现还是与人脑的能力无法相比。因此,可能需要考虑一种结合软件仿真的并行阵列混合方法。例如,如果每个处理器能够模拟1000个神经元的簇集,那么就可能更接近人脑的复杂程度。然而,具有10 000个芯片的系统必然价格昂贵。假设一个芯片从现在开始7年内花费1000美元,那么建立一个这样的系统将花费1千万美元。然而,随着晶体管价格每年下降约66.7%(Moore,2009),20年后,这样的系统将仅仅花费3000美元,但此类系统的功率损耗将仍然保持在20kW的水平上,因此除非能量的价格下降,否则该系统将因太昂贵而无法运行。

应该指出的是,上述讨论都是基于:假设摩尔定律将会继续适用于未来的数字芯片,与此同时,业界有很多关于法律对新技术不断发展的终极限制的讨论。从机器智能设计的角度来看,用硬件技术发展大规模综合智能系统能够显著增加社会利益。例如,FACETS(Fast Analog Computing with Emergent Transient States;2009)项目的目标是设计VLSI神经回路来模拟大脑的实质区域,他们最近设计的"芯片大脑"能够实现20万个神经元和5千万个突触,这为使用模拟VLSI技术设计大规模、复杂的智能系统提供了有力的支持。BioWall(2009)的研究是以可重构FPGA技术为基础的,向发展能够进化、自我修复、自我复制和学习的生物启发式电子组织迈出了重要的一步。BioWall项目的目标是发展真正的生物启发式硬件系统的机器智能,其范围是"从系统发育系统(受生物物种的进化启发)、通过个体发育系统(受多细胞生物的发展和生长启发),到后生系统(受个体对环境的适应启发)"。据报道,其涉及150~3500个单位的技术原型,它代表了大约3500个FPGA加上1个I/O接口的计算能力。

最后,我想讲述一个关于机器智能的硬件设计的故事。我参加过2009年在乔治亚州的亚特兰大举办的神经网络国际联合会议(IJCNN 2009),在会上我有幸听到了关于忆阻器("记忆电阻器"的简称)的最新发展报告。简单地说,忆阻器的概念是由Dr. Leon Chua在1971年提出的,他认为,忆阻器应该被认为是基于电阻器、电感器和电容器之间对称性的第4个基本电路元件(Chua,1971)。最近,惠普实验室宣布他们的团队已经建造了一个纳米级的开关忆阻器(Tour & He,2008;Strukov,Snider,Stewart & Williams,2008)。在机器智能领域中,有一个令人振奋的消息,即忆阻器可能显著地改善硬件设计能力,且可以用来开发能够模拟人类大脑复杂神经结构的集成电子电路。此外,由于忆阻器与神经元的突触以相似的方式处理电流和电压,也就是说,它们都可以在激活前使电压达到一个阈值并让电流通过,因此

使用忆阻器复制大脑神经信息处理具有独特优点。因此，新技术(如忆阻器)能够为我们提供所需的硬件平台，并带来更加接近于实际的类脑通用智能，当然只要我们知道如何使用它。由于对自然智能的理解以及发展集成化自适应系统来复制这一水平的智能仍然是最大的未解决的科学和工程挑战之一，我希望本书所提供的研究方法能够有助于这一具有挑战性、令人振奋的、有价值的研究领域的发展。

参考文献

Arden, W., Cogez, P., Graef, M., Ishiuchi, H., Osada, T., J. Moon, A. J. R., et al. (2005). International roadmap committee, international technology roadmap for semiconductors, executive summary.

Bayraktaroglu, I., Balkir, S., & Dundar, G. (1996). Annsis: A circuit level simulator for analog neural networks. *Turkish Symposium on Artificial Intelligence and Neural Networks*, pp. 305–310.

Bayraktaroglu, I., Ogrenci, S., Dundar, G., Balkir, S., & Alpaydin, E. (1997). Annsys: An analog neural network synthesis system. *Int. Conf. Neural Netw.*, pp. 910–915.

BioWall project [Online], Available: http://lslwww.epfl.ch/biowall/. (2009).

Chua, L. (1971). Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, *CT-18*(5), 507–519.

FACETS (fast analog computing with emergent transient states) project [Online], Available: http://facets.kip.uni-heidelberg.de/index.html. (2009).

Maly, W. (1996). The future of ic design, testing, and manufacturing. *IEEE Design and Test of Computers*, *13*(4), 8–91.

Mead, C. A. (1989a). Adaptive retina. In C. Mead & M. Ismail (Eds.), *Analog VLSI implementation of neural systems* (pp. 239–246). Norwell, MA: Kluwer Academic.

Mead, C. A. (1989b). *Analog VLSI and neural systems*. Reading, MA: Addison Wesley.

Mead, C. A. (1990). Neuromorphic electronic systems. *Proc. IEEE*, 78, 1629–1636.

Mead, C. A., & Ismael, M. (Eds.). (1989). *Analog VLSI implementation of neural systems*. Norwell, MA: Kluwer Academic.

Moore, G. E. (2009). Our revolution *[Online], Available: http://www.sia-online.org/galleries/default-file/Moore.pdf*.

Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, *453*, 80–83.

Tour, J. M., & He, T. (2008). Electronics: The fourth element. *Nature*, *453*, 42–43.

Vittoz, E. A. (1985). The design of high-performance analog circuits on digital CMOS chips. *IEEE J. Solid-State Circuits*, *20*(3), 657–665.

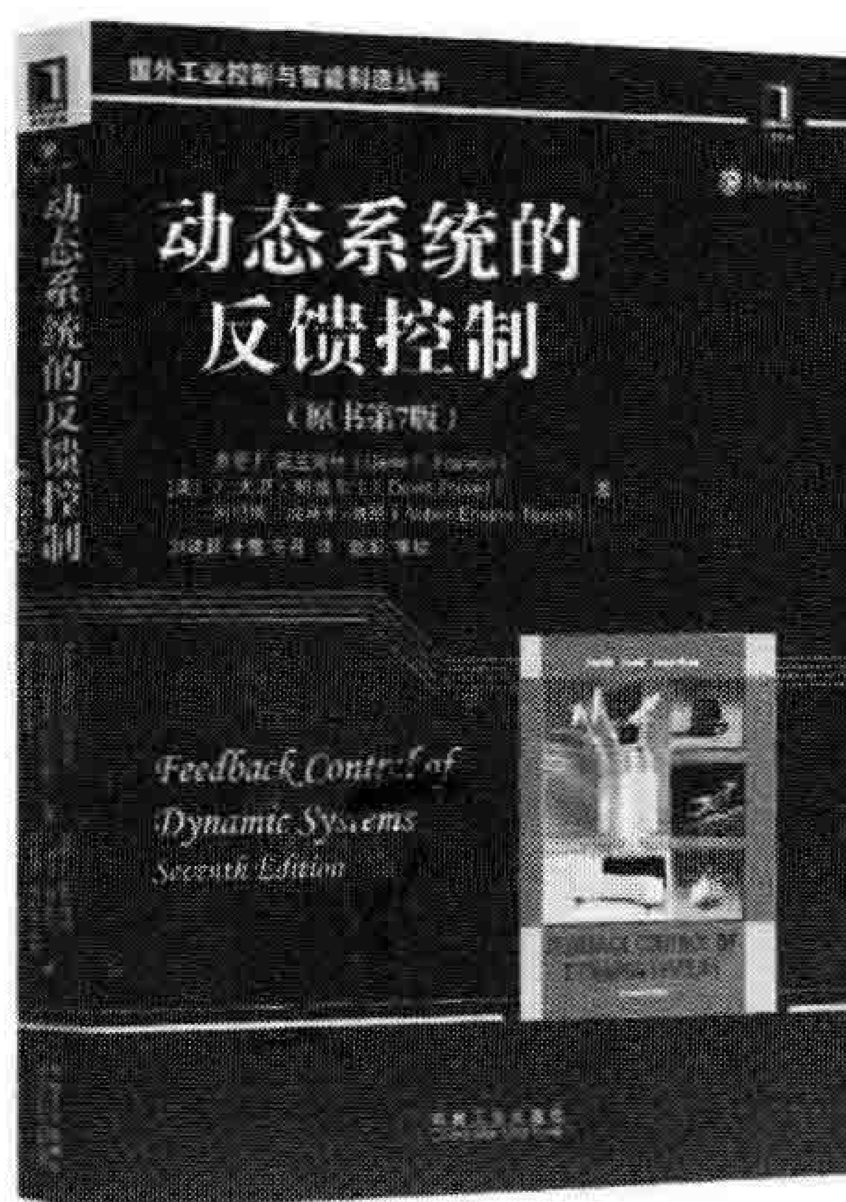
Vittoz, E. A. (1990a). Analog VLSI implementation of neural networks. *IEEE Int. Symp. Circuit and Systems*, 4, 2524–2527.

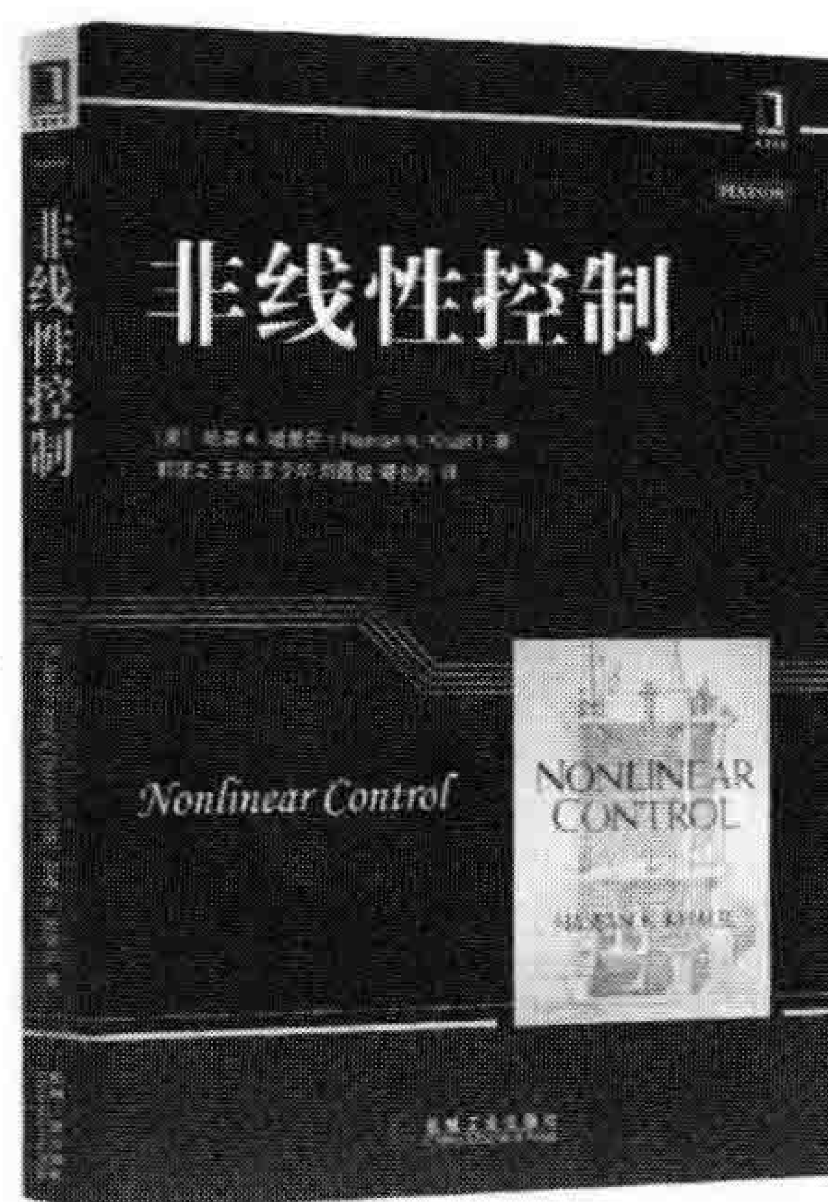
Vittoz, E. A. (1990b). Future of analog in the VLSI environment. *IEEE Int. Symp. Circuit and Systems*, 2, 1372–1375.

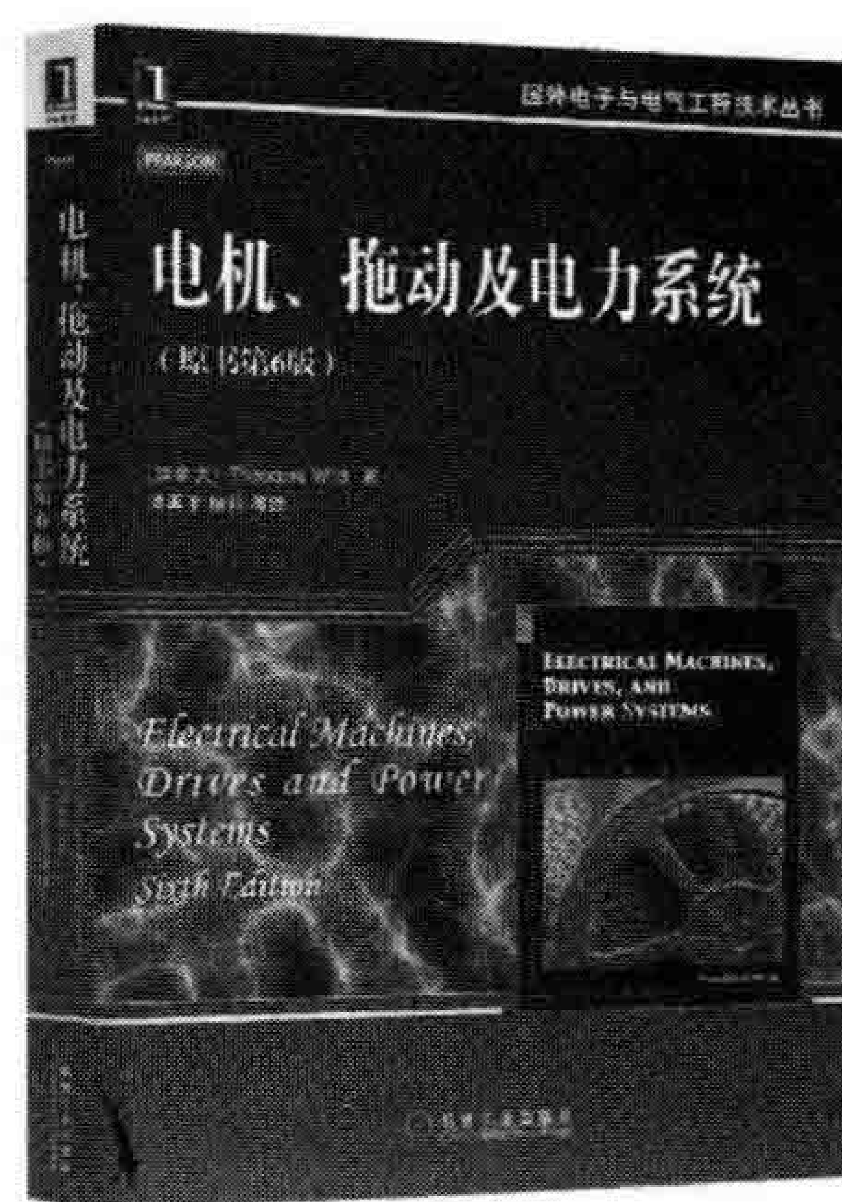
Vittoz, E. A. (1996). Biology inspired circuits. *IEEE Micro.*, *16*(5), 10.

Vittoz, E. A. (1998). Analog VLSI for collective computation. *IEEE Int. Conf. Electronics, Circuits and Systems*, 2, 3–6.

推荐阅读







动态系统的反馈控制（原书第7版）

作者：吉恩 F. 富兰克林 译者：刘建昌 等 ISBN：978-7-111-53875-2 定价：119.00元

本书系统阐述了反馈控制的基本理论、设计方法及在工程技术领域中的一些实际问题。本书主要是利用根轨迹、频率响应和状态变量方程这三种方法将控制系统的分析与设计结合起来，并结合大量实例和Matlab进行控制系统的分析与设计，来阐述相关内容。本版更加注重反馈控制在整个控制理论体系中的地位，因而对部分章节进行调整，对书中的实际例子进行了更新，而且对各章的习题进行修正和补充。

非线性控制

作者：哈森 K. 哈里尔 译者：韩正之 等 ISBN：978-7-111-52888-3 定价：79.00元

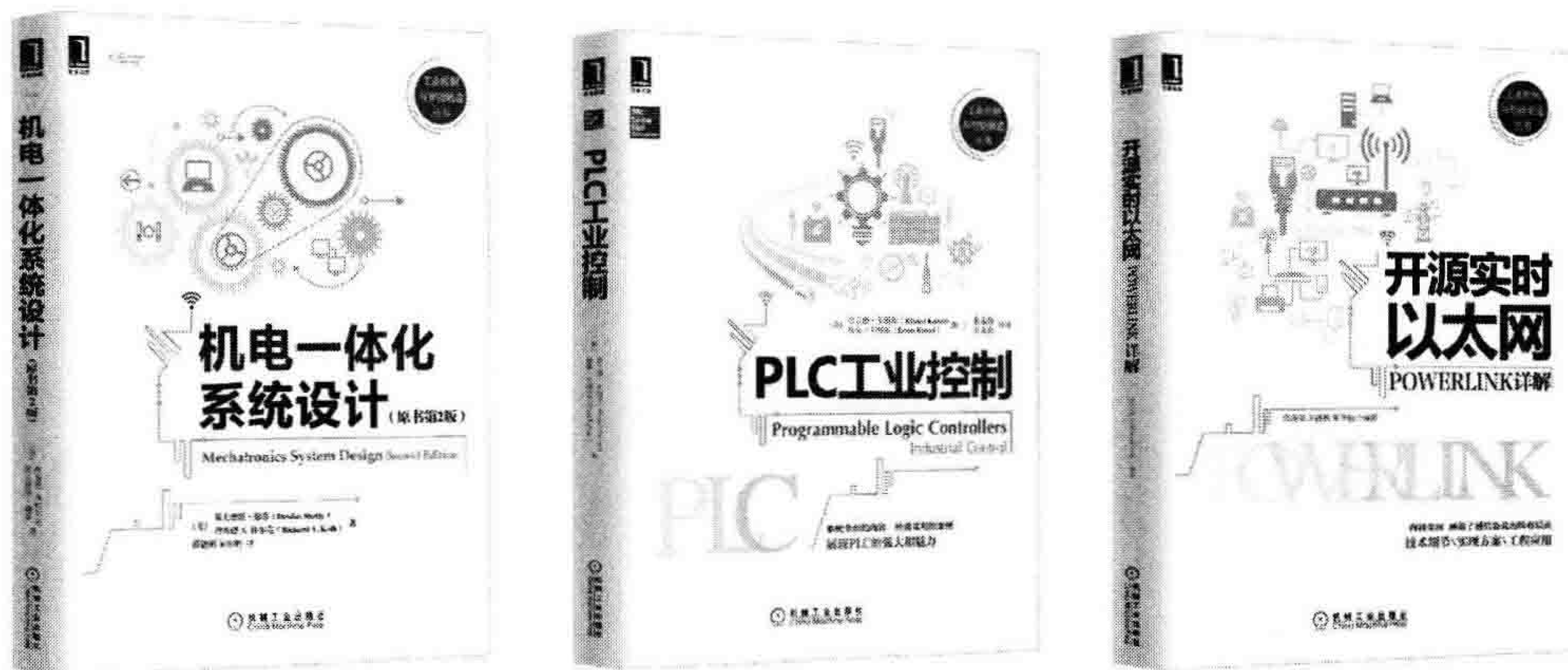
美国密歇根州立大学电气与计算机工程专业的本科生教材，可作为非线性控制课程的第一学期的教材，重点阐述非线性系统的分析和控制应用。该书内容简洁，阐述清楚，主要内容包括：非线性模型、两维系统、平衡点的稳定性、时变扰动系统、输入与输出的稳定性、稳定的反馈系统、特殊的非线性形式、状态反馈稳定性、鲁棒状态反馈稳定性等。全书共250多道习题，寓教于学，使用MATLAB和Simulink进行计算机仿真模拟，便于学生练习和掌握书的内容。

电机、拖动及电力系统（原书第6版）

作者：Sergio Franco ISBN：978-7-111-47471-5 出版时间：2015年1月 定价：99.00元

本书是电气工程领域的畅销教材，多方位地通过理论、实例分析为读者全面展示现代电力系统。主要包括电气工程中的电路原理、电机学、电力电子技术、电机控制、电力系统基础等课程的核心内容，分为四个部分：电气工程所需的电学、磁学、力学、热学及电路基本知识；直流电机、异步电机、同步电机及变压器等的基本原理；电力电子技术、直流电机与交流电机的电子控制等电气传动技术；最后涉及电力系统，包括新能源发电在内的各类发电厂、电能的传输与分配（包括直流输电）、电能的控制技术。本书适合作为电气类专业、非电气类专业人员学习或自学电气工程基础的教材与参考书。

推荐阅读



机电一体化系统设计（原书第2版）

作者：戴夫德斯·谢蒂 译者：薛建彬 ISBN：978-7-111-52923-1 定价：89.00元

本书深入讨论了机电一体化设计过程的关键内容，探讨了其发展方向，重点讲解系统建模和仿真，详细介绍了传感器和换能器的基本理论和概念、几种类型的驱动系统、控制和逻辑方法，特别是机电一体化系统中的控制设计，讨论了实时数据采集的理论和实践。最后还介绍了在智能制造领域机电一体化技术的发展。

PLC工业控制

作者：哈立德·卡梅 等 译者：朱永强 等 ISBN：978-7-111-50785-7 定价：69.00元

该书是一本介绍PLC编程的书，其关注点集中于实际的工业过程自动控制。全书以Slemens S7-1200 PLC的硬件配置和整体自动化集成（Totally Integrated Automation）界面为基础进行介绍讲解。其内容包括：自动化及过程控制基本概念、继电器逻辑基础及PLC结构和原理、PLC计数器和定时器的应用编程、模拟模块、梯形图逻辑和HMI、模块化程序设计、开环和闭环过程控制、综合性设计项目实例等。

开源实时以太网POWERLINK详解

作者：肖维荣 等 ISBN：978-7-111-50785-7 定价：49.00元

本书从现有工业以太网技术的比较开始，概要性的介绍了主流的几种实时以太网，详细介绍了POWERLINK实时以太网技术。内容包括POWRLINK的技术原理，特点，具有哪些功能，以及应用层CANopen的概念，这些基础的技术理论。进而介绍了如何实现和使用POWERLINK，包括如何组建和配置POWRELINK网络，如何诊断网络错误等。最后介绍了POWERLINK的典型应用，包括运动控制和过程控制。